CO759. Network Routing Game

Setup: Directed graph $G=(V,E)$

K players: each player is associated with an $s_i - t_i$ pair $s_i, t_i \in V$ and wants to route 1 unit of flow unsplittably from s_i to t_i

I.e. player i chooses a single path P_i from

$\mathcal{P}_i = \{P : P \text{ is an } s_i, t_i \text{ path}\}$ and routes its flow along P_i.

Each edge e has a latency/delay functions $l_e : \mathbb{R}_+ \to \mathbb{R}_+$

(will conventionally assume is monotonic increasing)

$l_e(x) \to$ delay on e given load x on e.

Given a path-selection $\{P_i\}$, let $f_e = |\{\#\text{ of players using } e\}|$

$= |\{i : e \in P_i\}|$ the cost of player i is the total delay of her path, i.e. $\ell_{P_i}(f) = \sum_{e \in P_i} l_e(f_e)$

Notation: Will say $(\{P_i\}, f)$ where f is congestion/load-vector associated with $\{P_i\}$

$(\{P_i\}, f)$ is a NE if $\forall i, \forall Q \in \mathcal{P}_i$, $\ell_{P_i}(f) \le \ell_Q(\hat{f})$

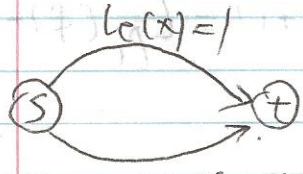
where $\hat{f}$ is congestion-vector associated with (Q, P_{-i})

$$\Leftrightarrow \hat{f}_e = \begin{cases} f_e - 1 & \text{if } e \in P_i \setminus Q \\ f_e & \text{if } e \in P_i \cap Q \\ f_e + 1 & \text{if } e \in Q \setminus P_i \end{cases} \Rightarrow (\{P_i\}, f) \text{ is an NE if}$$

$$\forall i, \forall Q \in \mathcal{P}_i, \quad \sum_{e \in P_i \setminus Q} l_e(f_e) \le \sum_{e \in Q \setminus P_i} l_e(f_e + 1) \qquad (*)$$

Examples:

$l_e(x) = 1$



$l_e(x) = \frac{x-2}{2}$

2 players each having $s_i = s$, $t_i = t$.

Unique NE is where both players choose bottom link.

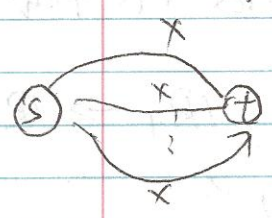
consider social-cost objective = sum of all player's costs,

$$C(f) = \sum_i \sum_{e \in P_i} l_e(f_e) = \sum_e f_e \, l_e(f_e)$$

$C(f^{NE}) \approx 2$

so The opt. flow sends 1 unit each on top and bottom paths, incurring cost $C(f^*) \approx 3/2$

Generalization of load-balancing game



$l_e(x) = x \ \forall e$ $\quad$ n players with $s_i = s$, $t_i = t$

$\equiv$ load balancing game

Does a pure NE exist?

Define the following potential function

$$\Phi(f) = \sum_e \sum_{i=1}^{f_e} l_e(i)$$

Claim: Given $(\{P_i\}, f)$ suppose player i switches to some $Q \in P_i$ to give $(\{Q, P_{-i}\}, \hat{f})$ Then

$$\Phi(f) - \Phi(\hat{f}) = l_{P_i}(f) - l_Q(\hat{f})$$

proof: $$\Phi(f) - \Phi(\hat{f}) = \sum_e \Big[\underbrace{\sum_{i=1}^{f_e} l_e(f_e) - \sum_{i=1}^{\hat{f}_e} l_e(i)}_{\Delta_e} \Big]$$

$\Delta_e = 0 \quad \forall e \in P_i \cap Q \quad \forall e \notin (P_i \cup Q)$

$\Delta_e = l_e(f_e) \quad \forall e \in P_i \setminus Q$

$\Delta_e = -l_e(f_e + 1) \quad \forall e \in Q \setminus P_i$

$$\Rightarrow \sum_e \Delta_e = \sum_{e \in P_i \setminus Q} l_e(f_e) - \sum_{e \in Q \setminus P_i} l_e(f_e + 1) = l_{P_i}(f) - l_Q(\hat{f})$$

$$\Phi(f) = \sum_e \sum_{i=1}^{f_e} l_e(i)$$

Theorem: Network routing game has a pure NE.

proof: Let $(\{P_i\}, f)$ be such that $\Phi(f)$ is minimum.

Then $\forall i, \forall Q \in P_i$

$$L_{P_i}(f) - L_Q(\hat{f}) = \Phi(f) - \Phi(\hat{f}) \leq 0$$

Φ is called an exact potential function.

Defn. Given a game $\mathcal{G} = (n, \{S_i\}, \{C_i\}_{i=1}^{n})$ $\mathcal{G}$ is called a potential game if there exists an (exact) potential function Φ for $\mathcal{G}$, i.e.

$\Phi : S_1 \times \cdots \times S_n \to \mathbb{R}$ s.t.

$\forall s \in S, \forall i \ \forall s_i' \in S_i, \ \Phi(s) - \Phi(s_i', s_{-i}) = C_i(s) - C_i(s_i', s_{-i})$

Thm: Suppose $\mathcal{G}$ is a potential game with pot. func. Φ Then

(i) $\mathcal{G}$ has a pure NE

(ii) If $s \in S$ is s.t. $\forall i, \forall s_i' \in S_i$

$\Phi(s) \leq \Phi(s_i', s_{-i})$ then s is a NE and vice versa.

(iii) Any sequence of improving moves gives a NE.

Thm. 2. holds even if Φ satisfies a weaker property that

$\forall s, \forall i, \forall s_i' \in S_i$

$$\Phi(s) - \Phi(s_i', s_{-i}) \geq 0 \iff C_i(s) - C_i(s_i', s_{-i}) \geq 0$$

such a Φ is called an ordinal potential func.

Price of Stability

Theorem 3: Suppose $\forall f$, we have $a \cdot C(f) \leq \Phi(f) \leq b \cdot C(f)$ for some constants a, b. Then $PoS \leq b/a$.

$$C(f) = \sum_i \sum_{e \in P_i} L_{P_i}(f) = \sum_e f_e \, l_e(f_e)$$

proof: Let f^{NE} be a minimizer of $\Phi(\cdot)$ and let f^* be an opt. soln. Then

$$C(f^{NE}) \leq \frac{\Phi(f^{NE})}{a} \leq \frac{1}{a} \cdot \Phi(f^*) \leq \frac{b}{a} C(f^*)$$

(Is true for even ordinal pot. func.

$\{\{P_i\}, f\}$ is a NE if

$$(*) \mid \forall i, \forall Q \in \mathcal{P}_i \quad \sum_{e \in P_i \setminus Q} l_e(f_e) \le \sum_{e \in Q \setminus P_i} l_e(f_e+1)$$

corollary: suppose all $l_e(\cdot)$ were of the form

(i) $a_e x + b_e$ (linear) $\Rightarrow PoS \le 2$

(ii) polynomials of degree p $\Rightarrow PoS \le p+1$

Recap. Bounds on PoA

Lemma: If $(\{P_i\}, f)$ is a NE then

$$C(f) = \sum_e f_e l_e(f_e) \le \sum_e g_e l_e(f_e+1) \text{ for any other}$$

congestion vector g obtained via a path-selection $\{Q_i\}$

proof: By $(*)$

$$\sum_{e \in P_i} l_e(f_e) \le \sum_{e \in P_i \cap Q_i} l_e(f_e) + \sum_{e \in Q_i \setminus P_i} l_e(f_e+1)$$

$$\le \sum_{e \in Q_i} l_e(f_e+1)$$

Adding this for all i gives

$$L(f) = \sum_i \sum_{e \in P_i} l_e(f_e) \le \sum_i \sum_{e \in Q_i} l_e(f_e+1)$$

$$= \sum_e l_e(f_e+1) \left| \{i : e \in Q_i\} \right| = \sum_e g_e l_e(f_e+1)$$

Atomic Routing Game:

Theorem: Suppose all $l_e(\cdot)$ are linear (ie) of the form $a_e x + b_e$) then $PoA \le 5/2$

proof: Let $\{\{P_i\}, f\}$ be a NE and let $(\{Q_i^*\}, f^*)$ be an opt. soln.

By above Lemma, $C(f) = \sum_e f_e l_e(f_e) \le \sum_e f_e^* l_e(f_e+1)$

$$= \sum_e f_e^* [a_e(f_e+1) + b_e]$$

suppose we could show that for $\forall e$.

$$f_e^* [a_e(f_e+1) + b_e] \le X \cdot f_e^* [a f_e^* + b_e] + Y \cdot f_e [a f_e + b_e]$$

where $\gamma < 1$, then

$\cancel{C(f^*)}$ $C(f) \leq \lambda \cdot C(f^*) + \gamma C(f) \Rightarrow$

$$(1-\gamma) C(f) \leq \lambda \cdot C(f^*) \Rightarrow C(f) \leq \frac{\lambda}{1-\gamma} C(f^*)$$

E.g.

$$f_e^* [a_e(f_e+1)+b_e] \leq f_e^* [2a f_e^* + b_e] + f_e^* a_e (b_e - f_e^*)]$$

$$\leq 2 \cdot f_e^* (a f_e^* + b_e) + \frac{1}{4} \cdot f_e - a_e b_e$$

$\Rightarrow$ can take $\lambda = 2$ $\gamma = \frac{1}{4}$

Claim: For all non-negative integers α, β

$$\alpha(\beta+1) \leq \frac{5}{3}\alpha^2 + \frac{1}{3}\beta^2$$

(1) $C(f) \leq \sum_e f_e^* [a_e(b_e+1)+b_e]$

$\Rightarrow$ $f_e^*[a_e(f_e+1)+b_e] = a_e f_e^* (f_e+1) + b_e f_e^*$

by claim

$$\leq a_e \left(\frac{5}{3} f_e^{*2} + \frac{1}{3} f_e^2\right] + b_e f_e^*$$

$$\leq \frac{5}{3} f_e^* (a f_e^* + b_e) + 1/3 f_e (a_e f_e + b_e))$$

Factors affecting PoA

- Crucially affected by latency funcs.
- PoA is affected by network Topology
  e.g. for load balancing games, s-t || link graph
  PoA for linear latencies is close to 2)
- How is PoA affected by locations of s-t pairs.

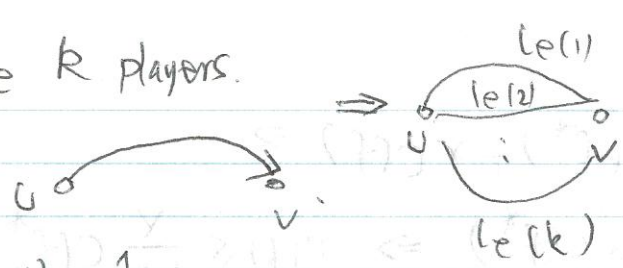
Computation of NE.

Consider $s_i = s$ $t_i = t$

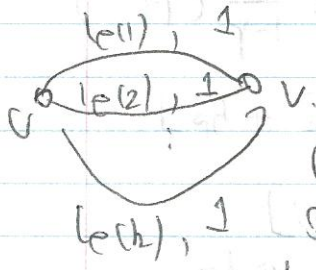
Theorem 3: A NE can be computed using a min-cost flow computation.

Proof: Recall $\Phi(f) = \sum_e \sum_{i=1}^{f_e} l_e(i)$

suppose k players.



capacity = 1



Solve min-cost flow problem where we want to send k units from $s \to t$.

① integer capacities $\Rightarrow$ integer min-cost flow.

② If min-cost flow routes, integer f_e units using 1 copies then it uses the first f_e copies.

Since cost $l_e(\ )$ is $\nearrow$ $\Rightarrow$ cost of integer min-cost flow $f = \phi(f)$

General case: Not Much is known.

Nonatomic Routing Games

SETUP: Directed graph G with latency functions $\{l_e(\ )\}$ on edges, where $l_e(\cdot) \nearrow$ continuous (and maybe differentiable)

k $s_i - t_i$ pairs. each $s_i - t_i$ pair has a certain total volume of traffic (demand r_i which we think as being composed of infinitely many players controlling an infinitesimal unit of flow. and this volumn is routed from $s_i \to t_i$.

A solution is a flow f st. $\forall i$

$$\sum_{p \in P_i} f_p = r_i$$

(Notation: will think of f as $(f_{e,i})_{e \in E,\ i=1,\dots,k}$ and its path-decomposition $\{f_p\}$

$$f_e \equiv \sum_{p \in P_i : e \in p} f_p \qquad f_p > 0 \quad \text{for } p \in P_i \equiv f_{e,i} > 0 \ \forall e \in p$$

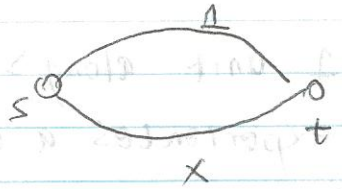
Define: f to be a Nash flow or NE if

(1) f is feasible

(2) $\forall i,\ \forall p \in P_i\ f_p > 0,\ \forall Q \in P_i \quad \underbrace{\sum_{e \in p} l_e(f_e)}_{= L_p(f)} \le \underbrace{\sum_{e \in Q} l_e(f_e)}_{= L_Q(f)}$

Ex.

(A) Pigou's example: $r_{st} = 1$ i.e. 1 unit flow from $s \to t$



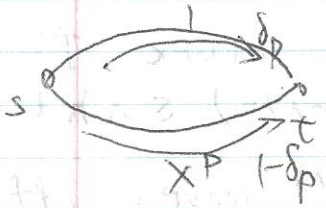
unique NE sends 1 unit of flow on bottom link.

Consider objective $C(f) = \sum_e f_e l_e(f_e) = \sum_p f_p l_p(f)$

Unique OPT sends 1/2 unit each on the two links $\Rightarrow$

$C(f^{NE}) = 1 \quad C(f^{OPT}) = 1 \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$

(B) Nonlinear Pigou's Example



$r_{st} = 1$. Again Unique NE sends 1 unit of flow on the bottom link.

$\Rightarrow C(f^{NE}) = 1$.

OPT: δ_p on top $1-\delta_p$ on bottom where $\delta_p \to 0$ as $p \to \infty$.

and incurs the cost $\to \frac{\ln p}{p}$ as $p \to \infty$

$\Rightarrow PoA \approx \frac{p}{\ln p} \to \infty$ as $p \to \infty$

May 21th 2008 Non atomic Routing Games:

Directed graph, G with continuous $l_e(\cdot)$ on edges, k, $s_i - t_i$ pairs each having a demand of r_i that is composed of infinitely many ε-users and which needs to be routed from s_i to t_i.

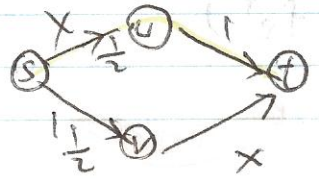
A flow f is a NE if.

(1) $\forall i \quad \sum_{p \in P_i} f_p = r_i$

(2) $\forall i \; \forall p \in P_i$ s.t. $f_p > 0 \quad \forall Q \in P_i \quad l_p(f) \le l_Q(f)$

Last time: Two Pigou-like examples.

(C) Braess' Example $\to$ NE and the optimum.

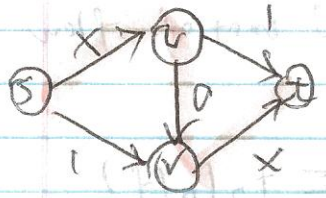


cost function is total cost

$= \sum_p f_p l_p(f) = \sum_e f_e l_e(f_e)$

Each ε-unit of flow experiences total delay $= 3/2$

$\Rightarrow C(f) = 3/2$.

$\Rightarrow$ 

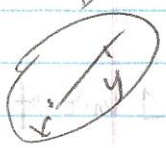
Unique NE sends 1 unit along $s \to u \to v \to t$ and every ε-user experiences a delay of of 2 units $\Rightarrow C(f) = 2$

Braess' Paradox.

FACTS from convex optimization

Defn: A set $D \subseteq \mathbb{R}^n$ is called convex if $\forall x, y \in D$

$\forall \lambda \in [0,1]$ $\lambda x + (1-\lambda) y \in D$



Defn: A function $h: \mathbb{R}^n \mapsto \mathbb{R}$ is convex if

$\forall x, y \in \mathbb{R}^n$ $\lambda \in [0,1]$ $h(\lambda x + (1-\lambda) y) \leq \lambda h(x) + (1-\lambda) h(y)$

if $h: \mathbb{R} \to \mathbb{R}$ and $h'(x)$ exists then h is convex iff $h'(x)$ is $\nearrow$ $\forall x$.

Theorem: Consider the following non-linear optimization problem

$$\text{(NLP)} \quad \min \sum_e h_e(f_e)$$

$$\text{st.} \quad f_e = \sum_i \sum_{p \in P_i : e \in p} f_p \qquad \text{where } h_e(\cdot) \text{ is convex differentiable } \forall e.$$

$$\forall i, \quad \sum_{p \in P_i} f_p = r_i$$

$$f_p \geq 0 \quad \forall p$$

Then the following are equivalent.

(i) f^* is an optimum soln. to (NLP)

(ii) $\forall$ feasible flows g to (NLP)

$$\sum_e h_e'(f_e^*) f_e^* \leq \sum_e h_e'(f_e^*) g_e$$

(iii) $\forall i, \forall p \in P_i$ st. $f_p^* > 0$ $\forall Q \in P_i$

$$\sum_{e \in p} h_e'(f_e^*) \leq \sum_{e \in Q} h_e'(f_e^*)$$

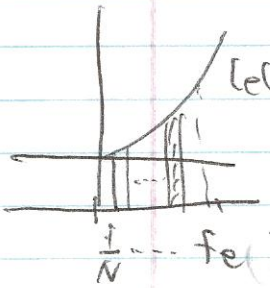
$\Phi(f) = \sum_e \int_0^{f_e} l_e(x)$

Moreover, f^{NE} is a NE iff f^{NE} minimizes

Theorem: A NE always exists in the nonatomic game. ·9·

proof: [Recall potential func. for atomic game

$$\Phi(f) = \sum_e \sum_{i=1}^{f_e} l_e(i) \ ]$$



$\Phi(\cdot)$ is of the form taking $h_e(f_e) = \int_0^{f_e} l_e(x)\,dx$

$h_e'(f_e) = l_e(f_e) \rightarrow$ is an $\nearrow$ fn.

so by part iii of theorem, f^{NE} minimizes $\Phi(\cdot)$

iff. $\forall i$, $\forall P \in P_i$ st. $f_P^{NE} > 0$ $\forall Q \in P_i$.

$$\sum_{e \in P} l_e(f_e^{NE}) \leq \sum_{e \in Q} l_e(f_e^{NE})$$

Theorem: Let f^{NE}, g^{NE} be two NE. Then $\forall e$ $l_e(f_e^{NE}) = l_e(g_e^{NE})$

proof: Take $h_e(f_e) = \int_0^{f_e} l_e(x)\,dx \Rightarrow h_e'(f_e) = l_e(f_e)$

f^{NE}, g^{NE} both minimize $\Phi(\cdot)$ $\Rightarrow$ By Theorem part (ii)

$$\sum_e l_e(f_e^{NE})(f_e^{NE} - g_e^{NE}) \leq 0 \qquad \sum_e l_e(g_e^{NE})(g_e^{NE} - f_e^{NE}) \leq 0$$

Adding $\Rightarrow$ $\sum_e (f_e^{NE} - g_e^{NE})(l_e(f_e^{NE}) - l_e(g_e^{NE})) \leq 0$

$\Rightarrow l_e(f_e^{NE}) = l_e(g_e^{NE})$

Corollary: If f^{NE}, g^{NE} are two NE's. Then $C(f^{NE}) = \sum_e f_e^{NE} \cdot l_e(f_e^{NE})$

$= C(g^{NE}) = \sum_e g_e^{NE} l_e(g_e^{NE})$.

proof: $C(f^{NE}) = \sum_i r_i L_i$ where $L_i = L_P(f^{NE})$ for any P st. $f_P^{NE} > 0$

L_i : does not change for g^{NE} $\Rightarrow C(g^{NE}) = \sum_i r_i L_i$

Consider the optimal flow f^* that minimizes $C(f)$

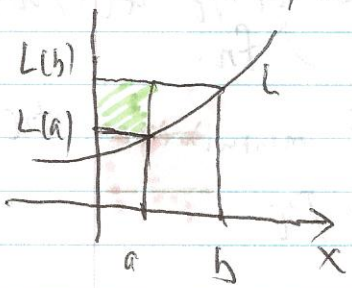
$C(f) = \sum_e f_e l_e(f_e) \rightarrow$ of the same form as obj'

function of (NLP) so if $h_e(f_e) = f_e \ell_e(f_e)$ is convex.

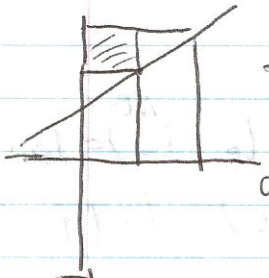
Bounds on PoA:

Defn: For a latency function: $\ell: \mathbb{R}_+ \to \mathbb{R}_+$ define

$$\alpha(\ell) = \max_{a,b \geq 0} \frac{b\ell(b)}{a\ell(a) + (b-a)\ell(b)} = \max_{a<b} \frac{b\ell(b)}{a\ell(a) + (b-a)\ell(b)}$$



Eg. If $\ell(x) = ax + b$.



then $\alpha(\ell) \leq 4/3$

Theorem: For a class $\mathcal{L}$ of latency functions, define $\alpha(\mathcal{L}) = \sup_{\ell \in \mathcal{L}} \alpha(\ell)$

Theorem: The worst case PoA for nonatomic instances (over all nonatomic routing instances) whose latency functions lie in class $\mathcal{L}$, where we assume $\mathcal{L}$ includes all constant latency functions, is exactly $\alpha(\mathcal{L})$

May 22th

For a latency func. $\ell: \mathbb{R}_+ \to \mathbb{R}_+$ define

$$\alpha(\ell) = \max_{a,b \geq 0} \frac{b\ell(b)}{a\ell(a) + (b-a)\ell(b)}$$

and for a class $\mathcal{L}$ of latency func. define $\alpha(\mathcal{L}) = \sup_{\ell \in \mathcal{L}} \alpha(\ell)$

Theorem: For any class $\mathcal{L}$ of latency func. containing all constant func. the worst-case PoA over all nonatomic instances introducing latency func's in $\mathcal{L}$ is $\alpha(\mathcal{L})$.

The worst case is attained on a Pigou-like example.

proof: upper bound of $\alpha(L)$: consider any instance with latency functions in $\mathcal{L}$, Let f^{NE} be NE, f^* be optimal flow

Then $C(f^{NE}) = \sum_e f_e^{NE} l_e(f_e^{NE}) \le \sum_e f_e^* l_e(f_e^{NE})$ (part ii) ...(*)

[Taking $h(f_e) = \int_0^{f_e} l_e(x)$].

[Recall in the atomic game, $\sum_e f_e^{NE} l_e(f_e^{NE}) \le \sum_e f_e^* l_e(f_e^{NE}+1)$]

suppose $\sum_e f_e^* l_e(f_e^{NE}+1) \le X \sum_e f_e^* l_e(f_e^*) + Y \sum_e f_e^{NE} l_e(f_e^{NE})$,

where $Y \le 1 \Rightarrow PoA \le \frac{X}{1-Y}$]

Suppose $\sum_e f_e^* l_e(f_e^{NE}) \le X \sum_e f_e^* l_e(f_e^*) + Y \sum_e f_e^{NE} l_e(f_e^{NE})$

then $PoA \le X/(1-Y)$

For any edge e. [taking $l = l_e$, $a = f_e^*$, $b = f_e^{NE}$)

we have

$$\alpha(L) \ge \alpha(l_e) \ge \frac{f_e^{NE} l_e(f_e^{NE})}{f_e^* l_e(f_e^*) + (f_e^{NE} - f_e^*) l_e(f_e^{NE})}$$

$\Rightarrow$

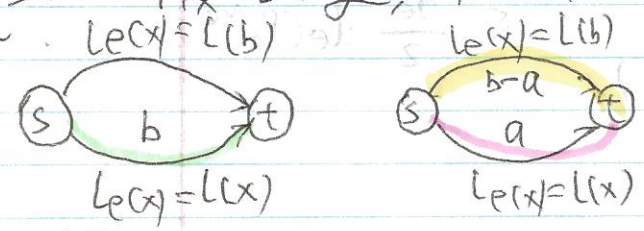
$$f_e^* l_e(f_e^{NE}) \le f_e^* l_e(f_e^*) + \left(1 - \frac{1}{\alpha(L)}\right) f_e^{NE} l_e(f_e^{NE})$$

$\Rightarrow$ in the $X = 1$ $Y = (1 - 1/\alpha(L))$] $\Rightarrow$

$PoA \le \alpha(L)$ [ie. $C(f^{NE}) \le \alpha(L) \cdot C(f^*)$]

Lower bound on PoA:

$r_{st} = b$ fix $L \in \mathcal{L}$, fix $b \ge 0$



NE:

$C(f^{NE}) = b \cdot l(b)$

for any $a \le b$ there is a flow $f^{(a)}$ that sends a unit on bottom link, $b-a$ on top $\Rightarrow C(f^{NE})/C(f^*) \ge C(f^{NE})/C(f^{(a)})$ for any $0 \le a \le b$

$$\Rightarrow \frac{C(f^{NE})}{C(f^*)} \geq \frac{b\,l(b)}{a\,l(a)+(b-a)\,l(b)} \quad \forall a,\ 0 \leq a \leq b$$

$$\Rightarrow \frac{C(f^{NE})}{C(f^*)} = \max_{0 \leq a \leq b} \frac{b\,l(b)}{a\,l(a)+(b-a)\,l(b)}$$

$\Rightarrow$ Maximizing over all $l \in \mathcal{L}$, all $b \geq 0$ gives $PoA \geq \alpha(\mathcal{L})$

what is $\alpha(\mathcal{L})$?

- For linear latencies, $\alpha(\mathcal{L}) = 4/3$
- For Polynomials of degree p with non negative coefficients.

$$\alpha(\mathcal{L}) = \frac{1}{1 - p/(p+1)^{1+\frac{1}{p}}} \rightarrow \frac{p}{\ln p} \text{ as } p \to \infty$$

Defn. Given an instance $\mathcal{I} = (G, \{l_e(\cdot)\}, \{(s_i, t_i, r_i)\})$ define $B\mathcal{I}$ to be the instance $(G, \{l_e(\cdot)\}, \{(s_i, t_i, Br_i)\})$

Theorem 2: Let f^{NE} be a NE for $\mathcal{I}$, and f^* be an optimal flow for $2\mathcal{I}$

Then $C(f^{NE}) \leq C(f^*)$

[In fact, for any $\delta \geq 0$, if $f^{(\delta)}$ is optimal flow for $(1+\delta)\mathcal{I}$, then $C(f^{NE}) \leq 1/\delta\, C(f^{(\delta)})$]

proof: $C(f^{NE}) = \sum_e f_e^{NE} l_e(f_e^{NE}) \leq \sum_e \frac{f_e^*}{2} l_e(f_e^{NE})$

(since $f^*/2$ is feasible for $\mathcal{I}$)

$$\Rightarrow C(f^{NE}) \leq \frac{1}{2}\sum_e f_e^* l_e(f_e^*) + \sum_e \underbrace{\frac{f_e^*}{2}\left[l_e(f_e^{NE}) - l_e(f_e^*)\right]}_{\leq \frac{f_e^{NE}}{2} l_e(f_e^{NE})}$$

$$\leq \frac{1}{2}C(f^*) + \frac{1}{2}C(f^{NE})$$

BICRITERIA BOUND

Ways of improving PoA.

1) Removing links from network: motivated from Braess Paradox
2) Resource Augmentation: Given Theorem 2, suppose one could come up with latency f'ns $\tilde{l}_e(\cdot)$ s.t

(i) f^{NE} is a NE wrt. $\tilde{l}_e(\cdot) \Rightarrow$
$\frac{f^{NE}}{2}$ is a NE wrt. $l_e(\cdot)$ and $\frac{r_i}{2}$ traffic.

(ii) $\tilde{C}(f^{NE}) :=$ cost of f^{NE} wrt. $\tilde{l}_e(\cdot) = C(\frac{f^{NE}}{2})$
cost of $\frac{f^{NE}}{2}$ wrt. $l_e(\cdot)$

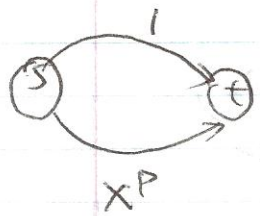
$\Rightarrow \tilde{C}(f^{NE}) = C(f^{NE}/2) \leq C(f^*)$ optimum for r_i instance wrt. $l_e(\cdot)$

E.g. $\tilde{l}_e(x) = \frac{1}{2} l_e(\frac{x}{2})$ — verify.

e.g. $l_e(x) = \frac{1}{v_e - x} \rightarrow$ arises in queuing theory applications.

then $\tilde{l}_e(x) = \frac{1}{2v_e - x} \Rightarrow$ blowing up capacities by 2 completely mitigates inefficiency.

3) Controlling a "small" δ-traffic, fraction of stackelberg strategies.



Open: Whether controlling δ-traffic helps in general graphs with one s-t pair.

However, in s-t $\|^l$ link graphs $PoA \leq 1/\delta$.

4) Imposing Tolls / Taxes on the edges:

Imposing certain tolls (τ_e) on edges, changes $l_e(\cdot)$ to $\tilde{l}_e(\cdot)$ where $\tilde{l}_e(x) = l_e(x) + \tau_e$

For any network, (any latency f'ns, $\{l_e(\cdot)\}$.)
$\exists\, \tau_e$ st. NE wrt $\tilde{l}_e(\cdot)$ is an optimal flow wrt $l_e(\cdot)$. § 18.5.1.

May 27.

Recap. Theorem Consider (IVLP) $\min \sum_e h_e(f_e)$
st. $\forall e \quad \sum_i \sum_{P \in P_i} f_P = f_e$

$\forall i \quad \sum_{p \in P_i} f_p = r_i$ where $f_e(\ )$ is convex, differentiable $\forall e$.

$f_p \geq 0 \quad \forall p$.

Then the following are equivalent.

(i) f^* is an opt. soln to (NLP)

(ii) $\forall g$ feasible to (NLP) $\sum_e h_e'(f_e^*)(f_e^* - g_e) \leq 0$

(iii) $\forall i, \forall p \in P_i$ st. $f_p^* > 0$ $\forall Q \in P_i$, $\sum_{e \in p} h_e'(f_e^*) \leq \sum_{e \in Q} h_e(f_e^*)$

proof: (i) $\Rightarrow$ (ii) Since f^*, g are feasible for any $\varepsilon > 0$.

$f_e = f_e^* + \varepsilon(g_e - f_e^*) = (1-\varepsilon) f_e^* + \varepsilon g_e$ is also feasible to (NLP)

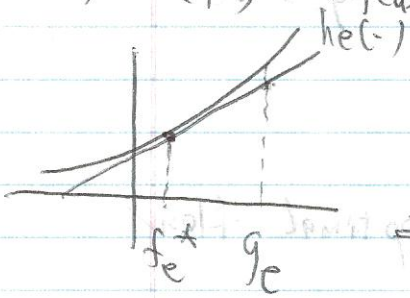
can write $h_e(f_e) = h_e(f_e^*) + \varepsilon h_e'(f_e^*) \underbrace{\Delta_e}_{g_e - f_e^*} + O(\varepsilon^2)$

$$\Rightarrow \sum_e h_e(f_e) = \sum_e h_e(f_e^*) + \varepsilon \underbrace{\sum_e h_e'(f_e^*)\Delta_e + O(\varepsilon^2)}$$

If $\sum_e h_e'(f_e^*)\Delta_e < 0$, then for $\varepsilon > 0$ small enough, we have $\sum_e h_e(f_e) < \sum_e h_e(f_e^*)$ — contradiction.

[so $\sum_e h_e'(f_e^*)(g_e - f_e^*) \geq 0$]

(ii) $\Rightarrow$ (i) $\forall$ feasible soln g to (NLP)



Then have

because of convexity of $h_e(\cdot)$)

$h_e(g_e) \geq h_e(f_e^*) + h_e'(f_e^*)(g_e - f_e^*)$

$$\Rightarrow \sum_e h_e(g_e) \geq \sum_e h_e(f_e^*) + \underbrace{\sum_e h_e'(f_e^*)(g_e - f_e^*)}_{\geq 0}$$

$\Rightarrow f^*$ is an optimal soln. to (NLP)

(ii) $\Leftrightarrow$ (iii) exercise.

CONGESTION GAMES : Generalize routing games

A congestion game is given by $G = (N, E, \{P_i\}_{i=1}^{|N|}, \{l_e\}_{e \in E})$

where – N is a set of k players $\{1, \dots k\}$

– E is a set of m resources.

– $P_i \subseteq 2^E$ → family of subsets of E denoting the strategy set of $i \in N$.

– $l_e(\cdot)$ is the latency f'n. of resource $e \in E$

$l_e(x) \equiv$ delay on resource e if x players "use" e.

The cost to a player i under a strategy profile $(P_1, \dots P_k)$ is given by $\sum_{e \in P_i} l_e(f_e)$ where

$f_e = |\{ j \in N : e \in P_j \}|$

NETWORK CREATION GAMES.

SETUP : set V of n players. each node/player u may choose to build edges to any subset of the other nodes.

Formally, the strategy S_u of u is simply a subset of $V \setminus \{u\}$.

A strategy profile $S = (S_1, \dots S_n)$ gives rise to an undirected graph $G = G(S) = (V, E)$ where

$E = \{ (u,v) : u \in S_v, \text{ or } v \in S_u \text{ (or both)} \}$

unordered

≥ 0 parameter

building cost

The cost to player u under S is denoted by distance cost.

$$\text{cost}_u(S) = \alpha |S_u| + \sum_{v \neq u} D(u,v) \text{ where}$$

$D(u,v)$ = shortest hop-distance from u to v

$\equiv$ least # of edges on a u–v path.

∞ if u, v disconnected

Optimal

Objective function and optimal graphs

Define total-cost objective (social) $SC(G(S)) = \sum_u cost_u(S)$

In any NE, any edge (u,v) of G is bought by exactly one of u and v $\Rightarrow$ sets S_u are disjoint.

$$\Rightarrow SC(G) = \alpha \sum_u |S_u| + \sum_{u,v\ v \neq v} D(u,v)$$

$$= \alpha |E| + \sum_{u,v: u \neq v} D(u,v)$$

Theorem 1: if $\alpha \leq 2$: Complete graph is an optimal graph

$\alpha \geq 2$: Star is an optimal soln.

Proof sketch: $SC(G) \geq \alpha |E| + \sum_u (1 \cdot d_u + 2(n-1-d_u))$

$\alpha \geq 2$ $\quad d_u := \deg$ of u

$$= (\alpha - 2)|E| + 2n(n-1)$$

Equality holds when $diam(G) = 2$

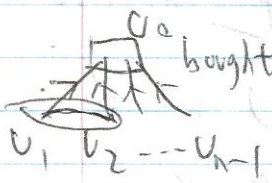
$\Rightarrow \alpha \leq 2$, want to maximize $|E|$ $\Rightarrow$ complete graph

$\alpha \geq 2$ - - - - minimize $|E|$ $\Rightarrow$ star but connected

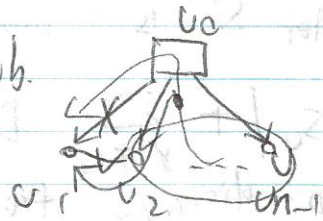
Pure NE: Lemma 2: If $\alpha \geq 1$, then a star is a NE where center buys all edge)

$\alpha \leq 1$, then $[cost_u(S) = \alpha |S_u| + \sum_{v \neq u} D(u,v)]$

Complete graph is a NE.



bought by the hub.



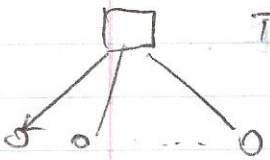
May 29th,

Recap. Theorem 1: $\alpha \geq 2$ star is an opt. soln.

$\alpha \leq 2$: Complete graph is an opt. soln.

Lemma 2 $\alpha \geq 1$ star is a NE

$\alpha \leq 1$ - complete graph is a NE.



Theorem 3: For $\alpha \leq 1$, $\alpha \geq 2$ $PoS = 1$

For $\alpha \in (1,2)$, $PoS \leq 4/3$

proof: For $\alpha \in (1,2)$ since star is a NE.

$$PoS \leq \frac{sc(\text{star})}{sc(\text{complete})} \leq 4/3$$

PoA bounds:

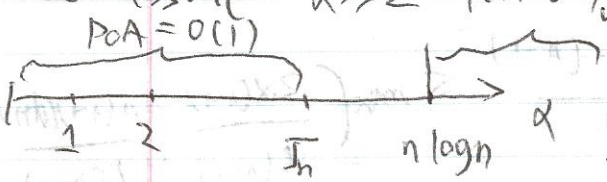
For $\alpha < 1$ complete graph is the unique NE $\Rightarrow$

$PoA = 1$

$\alpha \leq 2 \rightarrow$ every NE must have diametor ≤ 2

$\Rightarrow$ star is worst NE for $\alpha \in (1,2)$ $\Rightarrow PoA \leq 4/3$

so assume $\alpha \geq 2$ $PoA = O(1)$ will show that



— for any α $PoA = O(\sqrt{\alpha})$

— For $\alpha = O(\sqrt{n})$ $PoA = O(1)$

Also known that

OPEN: Is $PoA = O(1)$ for all α?

Lemma 4: Let G be a NE (for some α)

Then $\text{diam}(G) \leq 2\sqrt{\alpha}$.

proof: Consider any two nodes $u, v \in V$

Let $D(u,v) = L \geq 2k$ (for some k)

suppose u buys edge (u,v) $\Rightarrow$ building cost $\nearrow \alpha$

Reduction in distance cost $\geq (l-1)+(l-3)+\cdots+1$

$$\geq 2k-1+2k-3+\cdots+1 = k^2$$

If $D(u,v) > 2\sqrt{\alpha}$, then redn. in distance cost $> \alpha$

$\Rightarrow$ better to buy (u,v) $\Rightarrow$ G is not a NE.

~~Lemma~~ Theorem: If G is a NE with diam (d), then

$$SC(G) \leq (d+1)\,OPT.$$

Claim: Let G be a NE. Then for any $u \in V$, we have

$$SC(G) \leq 2\alpha(n-1) + n\sum_{v \neq u} D(u,v) + (n-1)^2$$

proof of Theorem (assuming claim)

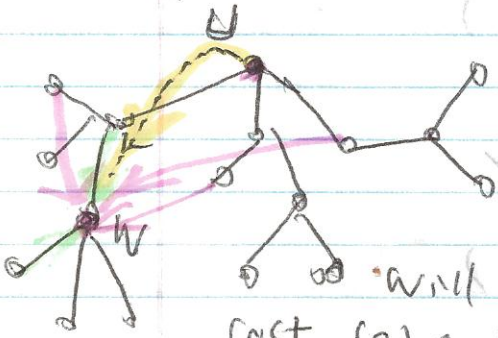
pick any node $u \in V$.

By claim, $SC(G) \leq 2\alpha(n-1) + n\cdot(n-1)\cdot d + (n-1)^2$

Also $OPT \geq \alpha(n-1) + n(n-1)$

$$\Rightarrow \frac{SC(G)}{OPT} \leq \frac{2\alpha(n-1) + n(n-1)d + (n-1)^2}{\alpha(n-1) + n(n-1)} \leq \max\left(\frac{2\alpha(n-1)}{\alpha(n-1)}, \frac{n(n-1)(d+1)}{n(n-1)}\right) = d+1.$$

proof of claim: Consider a BFS-tree T rooted at u. S: strategy profile s.t. $G = G(S)$



Let E_v = # of tree edges that v buys.

$$cost_u := \alpha|E_u| + \sum_{v \neq u} D(u,v) \quad \cdots (1)$$

Will show that $\forall v \neq u$

$$cost_w(S) \leq \alpha(E_w+1) + \sum_{v \neq u} D(u,v) + (n-1) \quad \cdots (2)$$

[$\Rightarrow$ Adding (1)+(2) $\Rightarrow$ for all $w \neq u$ gives

$$SC(G) \le \alpha(|T| + n-1) + n \sum_{v \neq u} D(u,v) + (n-1)^2]$$

Suppose w changes its strategy so that
- it buys all its tree edges
- deletes all its non-tree edges
- buys edge (u,w)

$$\text{cost}_w(S) \underset{\because NE}{\le} \text{cost}_w(\text{new strategy as above}) \le \alpha(E_w+1) + \sum_{v \neq w} D(w,v)$$

$$\le \alpha(E_w+1) + \sum_{v \neq w,u} (D(u,v)+1)+1$$

$$\le \alpha(E_w+1) + \sum_{v \neq u} D(u,v) + (n-1)$$

Corollary: $PoA = O(\sqrt{\alpha})$

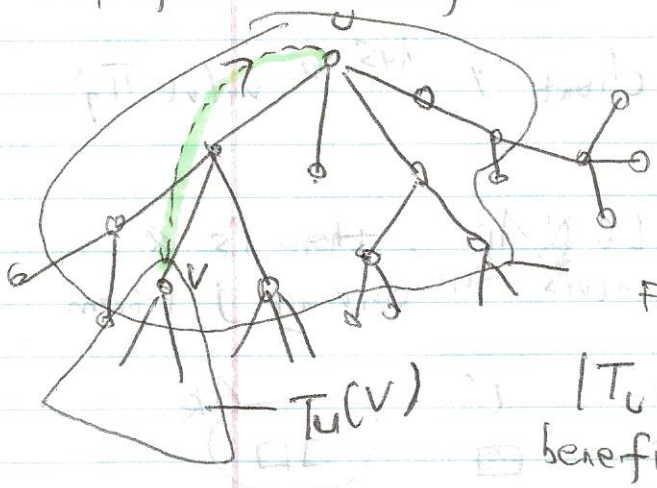
Now consider $\alpha \le \sqrt{n}$

Theorem 6: Let G be a NE (for $\alpha \le \sqrt{n}$)

Then $\text{diam}(G) \le 5 \Rightarrow SC(G) \le 6\, OPT$.

$\Rightarrow PoA \le 6$.

proof: Consider any node u, let $N_2(u) = \{v : D(u,v) \le 2\}$



Let D_i = # of nodes at distance i from u

$|N_2(u)| = 1 + D_1 + D_2$.

For every node v st $D(u,v)=2$.

$|T_u(v)| \le \alpha$ ow. would be beneficial to buy (u,v)

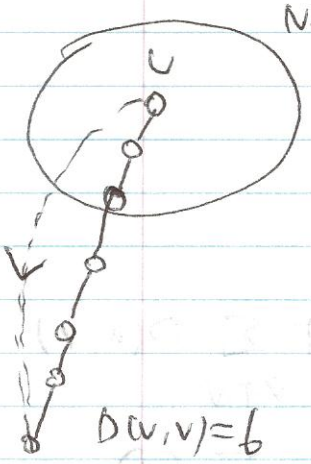
$$\Rightarrow n = 1 + D_1 + \sum_{v: D(u,v)=2} |T_u(v)| \le 1 + D_1 + \sum_{v: D(u,v)=2} \alpha$$

$$= 1 + D_1 + \alpha D_2$$

$$\Rightarrow |N_2(v)| = 1 + D_1 + D_2 > n/\alpha \quad [\alpha > 1]$$

$|N_2(u)| = 1 + p_1 + p_2 \geq n/\alpha \quad [\alpha \geq 1]$

suppose $\text{diam}(G) \geq 6$ let u, v be such $D(u,v) = 6$



$N_2(u)$ if v buys edge (u,v) then it reduces its distance to all nodes in $N_2(u)$ by ≥ 1

$\Rightarrow |N_2(u)| \leq \alpha$

$\frac{n}{\alpha}$

contradiction since $\alpha \leq \sqrt{n}$

June 3rd 2008 CO759 Algorithmic Game Theory

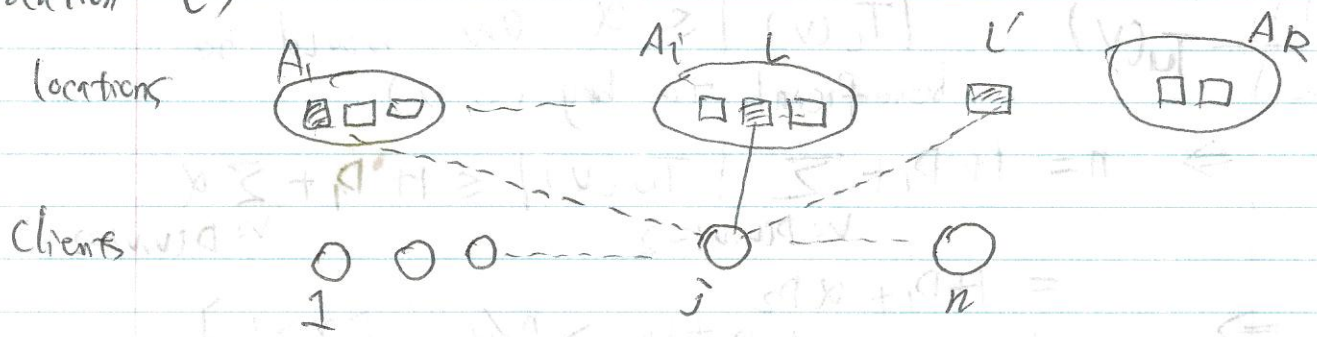
Facility Location Game.

Setup: k firms which are the players each player i has a disjoint set A_i of locations at which it can build facilities.

its strategy-set is $\mathcal{A}_i \subseteq 2^{A_i}$ for simplicity will assume that a firm can build a facility at most one location in A_i, i.e.

$$\mathcal{A}_i = \{\emptyset\} \cup \bigcup_{l \in A_i} \{\{l\}\}$$

- n clients/customers/markets, each client j has a value π_j for being served by a facility.
- for each client j and each location $l \in A_i$, there is a connection-cost c_{lj} that the firm i incurs for serving j (from location l)



Assume WLOG. $C_{lj} \le \pi_j \quad \forall l, j$

suppose i builds facility at $l \in A_i$ which is used to service client j and charges price p to j.

— $p \ge C_{lj}$ [otherwise i has no benefit in serving j]

— $\pi_j \ge p$ [- - - - - j has - - - - from service]

suppose i' has a facility at location $l' \in A_{i'}$ and proposes a price p' [$C_{l'j} \le p' \le \pi_j$] to j, then j would want to get served by l (and i) if

if $p \le p'$ → for every feasible price p' that some other firm i' could propose to j.

⇒ Given a strategy profile $(S_1, \dots S_k)$ [where each $S_i = \emptyset$ or some locations $l \in A_i$]

— ~~each~~ let $L = \bigcup_i S_i$ → locations at which facilities are built]

— each j is assigned to the location $l \in L$ st. $C_{lj} = \min_{l' \in L} C_{l'j}$

[or assigned to i st. $C_{s_i j} = \min_{i'} C_{s_{i'} j}$]

— price that i (st. $C_{s_i j} = \min_{i'} C_{s_{i'} j}$) charges to j is $\min_{i' \ne i} C_{s_{i'} j} = \min_{l' \in L,\, l' \ne s_i} C_{l'j}$ → second price game or VICKREY pricing scheme.

— utility $U_i(S) = \sum_{j:\, C_{s_i j} = \min_{i'} C_{s_{i'} j}} (p_{ij} - C_{s_i j})$

$$= \sum_{j:\, C_{s_i j} = \min_{i'} C_{s_{i'} j}} \left[\left(\min_{i' \ne i} C_{s_{i'} j} \right) - C_{s_i j} \right]$$

Objective function Social Welfare objective function

$V(s) \equiv$ Total value received by Society

$$= \sum_j \pi_j + \sum_i \Big(-\sum_{j: C_{s_i j} = \min_{i'} C_{s_{i'} j}} C_{s_i j}\Big)$$

$$= \sum_j \Big(\pi_j - \min_{i'} C_{s_{i'} j}\Big)$$

If i serves j under S ($C_{s_i j} = \min_{i'} C_{s_{i'} j}$)

and charges price p to j

Then i's utility via j $= p - C_{s_i j}$

$\oplus$

and j's benefit $\oplus = \pi_j - p$

$\pi_j - C_{s_i j}$

Theorem 1:

The Facility Location game is a potential game with potential function $V(\cdot) \Rightarrow$ by defn. $PoS = 1$

proof: $U_i(s_i, s_{-i}) - U_i(s_i', s_{-i}) = V(s_i, s_{-i}) - V(s_i', s_{-i})$

suffices to show $U_i(s) = V_i(s_i, s_{-i}) - V(\phi, s_{-i})$

RHS: let $C_i = \{j : C_{s_i j} = \min_{i' \neq i} C_{s_{i'} j}\}$ clients served by i

$$RHS = \Big[\sum_{j \in C_i} (\pi_j - C_{s_i j}) + \sum_{j \notin C_i} (\pi_j - \min_{i' \neq i} C_{s_{i'} j})\Big]$$

$$= \Big[\sum_{j \in C_i} (\pi_j - \min_{i' \neq i} C_{s_{i'} j}) + \dots\Big]$$

$$= \sum_{j \in C_i} (\min_{i' \neq i} C_{s_{i'} j} - C_{s_i j}) = U_i(s)$$

Utility Games: generalizes FL game.

A utility game consists of

— k players ; each player i is associated with a disjoint set A_i

Define $A = \bigcup_i A_i$

and a strategy set $\mathcal{A}_i \subseteq 2^{A_i}$

— each i has a utility function $U_i : 2^A \to \mathbb{R}$ (cautious)

Also here a SW function $V : 2^A \to \mathbb{R}$

The U_i's and V must satisfy

(1) $V(\cdot)$ is submodular, i.e.

[Defn: $\forall S \subseteq T \subseteq A \quad \forall L \notin T$

$V(T \cup \{L\}) - V(T) \leq V(S \cup \{L\}) - V(S)$]

(2) $\sum_i U_i(S) \leq V(S) \quad \forall S$

(3) $\sum_i U_i(S) \geq V(S) - \underline{V(\emptyset, S_{-i})} \; (= V(S_{-i}))$

If (3) is satisfied at equality $\to$ basic utility game

If $V(S) \leq V(T) \quad \forall S \subseteq T \to$ monotone utility game

Theorem 2: FL game is a basic monotone utility game. (exercise)

Theorem 3: A basic utility game is a potential game with potential function $V(\cdot) \Rightarrow PoS = 1$

Theorem 4: If Suppose a monotone utility game has a pure NE S. Then $V(S) \geq \frac{1}{2} OPT$ where $OPT = \max_{S'} V(S')$

proof: let $O^* = (O_1^*, \cdots O_k^*)$ be an optimal solution.

$V(O^*) - V(S) \leq V(O^* \cup S) - V(S) \quad (*)$

by monotonicity

Let $O^{(0)} = \emptyset \quad O^{(i)} = \bigcup_{j \leq i} O_j^*$ for $i = 1, \cdots k$.

$V(O^* \cup S) - V(S) \leq \sum_{j=1}^{k} V(O^{(j)} \cup S) - V(O^{(j-1)} \cup S)]$

$= \sum_{j=1}^{k} [V(O_j^* \cup (O^{(j-1)} \cup S)) - V(O^{(j-1)} \cup S)]$

— submodularity.

$$\le \sum_{j=1}^{k} [\, V(O_j^* \cup S_{-j}) - V(S_{-j}) \,]$$

$$\le \sum_{j=1}^{k} U_i(O_j^*, S_{-j}) \le \sum_{j=1}^{k} U_i(\cancel{O_j}\, S_j, S_{-j}) \le V(S)$$

since NE — 2nd property.

June 5th 2008. C0759 Existence of (mixed) NE

Recall Defn. A (simultaneous-move) game consists of k players, each player i having a strategy-set S_i and a payoff / utility function $U_i : S_1 \times \cdots \times S_k \to \mathbb{R}$

Notation: $S = S_1 \times S_2 \cdots \times S_k$ let $n_i = |S_i|$ and define $\Delta_n = \{ x \in \mathbb{R}^n : x_i \ge 0 \ \forall i \ \sum_{i=1}^{n} x_i = 1 \}$

A mixed-strategy profile is a tuple $x = (x^{(1)}, \ldots x^{(k)})$ where each $x^{(i)} \in \Delta_{n_i}$. A mixed profile x is a mixed NE if $\forall i$, $\forall \bar{x} \in \Delta_{n_i}$, $E_{s \sim x}[U_i(s)] \ge E_{s \sim (\bar{x}, x^{(-i)})}[U_i(s)]$

Expected value of $U_i(s)$ when each player j choose $s_j \in S_j$ with prob. $x_{s_j}^{(j)}$

Nash's Theorem: Every finite game has a mixed NE.

Brouwer's fixed pt. (FP) Theorem: let $f : D \to D$ be a continuous function where $D \subseteq \mathbb{R}^n$ is a closed, bounded and convex set. Then $\exists x \in D$ st. $f(x) = x$ — called a fixed pt. of f.

All conditions are essential.

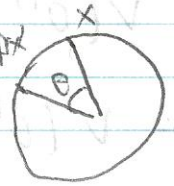
— Continuous:

— Closed: $D = (0, 1) \to f(x) = x/2$

— Bounded:

— Convex: $D = [-1, -\frac{1}{4}] \cup [1/4, 1]$ $f(x) = -x$

D = Circumference



f: rotation.

Proof of Nash's Thm using Brouwer : let $n = \sum_{i=1}^{k} n_i$, $\Delta = \Delta_{n_1} \times \cdots \times \Delta_{n_k}$

IDEA: f maps mixed profiles to mixed profiles st. if $x \in \Delta$ is st. some player j has an improving mixed strategy, then $f(x)$ "moves" towards that improving mixed strategy.

For example, given for each $x = (x^{(1)}, \cdots x^{(k)}) \in \Delta$, we could fix some improving strategy $\bar{x}^{(i)}$ for i.

[i.e. $E_{s \sim (\bar{x}^{(i)}, x^{(-i)})}[U_i(s)] \geq E_{s \sim x}[U_i(s)]$ and set

$f(x) = (\bar{x}^{(1)}, \cdots \bar{x}^{(k)})$

with strict inequality if $\exists$ a strictly improving strategy for i.

BUT f is NOT garanteed to be continuous.

To ensure continuity, given $x \in \Delta$, $\forall i$, $\forall t \in S_i$

define $d_{it}(x) = E_{s_{-i} \sim x^{(-i)}}[U_i(t, s_{-i})] - E_{s \sim x}[U_i(s)]$

continuous.

improvement when i plays pure strategy t.

— Define $f(x) = (y^{(1)}, \cdots y^{(k)})$ where

$y_t^{(i)} = x_t^{(i)} + d_{it}$ $\longrightarrow$ $\uparrow$ prob. on strategy t where $d_{it} > 0$

— $y_t^{(i)}$ could be < 0 ($\because d_{it} < 0$) $\downarrow$ prob. on strategies where $d_{it} < 0$

— $\sum y_t^{(i)}$ need not be 1

$$y_t^{(i)} = \frac{x_t^{(i)} + \max(d_{it}, 0)}{\sum_{t \in S_i}(x_t^{(i)} + \max(d_{it}, 0))} \quad (\geq 0)$$

— normalize

$f(x) = y$ where $y_t^{(i)} = \dfrac{x_t^{(i)} + \max(d_{it}, 0)}{\sum_{t \in S_i}(x_t^{(i)} + \max(d_{it}, 0))}$

By Brouwer $\exists x^*$ st. $x^* = f(x^*)$ $\rightarrow$ x^* must be a NE.

$\rightarrow$ I.e. $d_{it}(x^*) \leq 0$

Note that $\sum_{t \in S_i} x_t^{*(i)} d_{it}(x^*) = 0$ — for any x

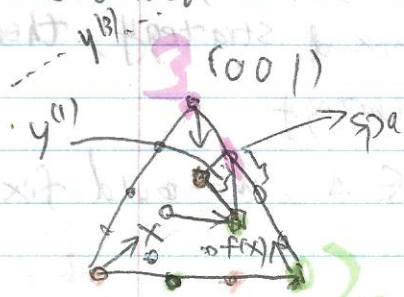
since

Since $x^* = f(x^*)$, $\left.\begin{array}{l} d_{it}(x^*) > 0 \Rightarrow x_t^{*(i)} > 0 \end{array}\right\} \Rightarrow d_{it}(x^*) \le 0 \quad \forall i, t \in S_i$

see slides on the web.

Sperner's Lemma

Brouwer when $D = \Delta_3 = \{ x \in \mathbb{R}^3 : \sum_{i=1}^{3} x_i = 1 \quad x_i \ge 0 \ \forall i \}$



$x = (\lambda_1, \lambda_2, \lambda_3)$

$f(x) = (\mu_1, \mu_2, \mu_3)$

$x \Leftrightarrow$ colour i where i is the smallest index st. $\lambda_i > \mu_i \equiv x_i > f(x)_i$

$x = (a_2, a_3, c)$

$f(x) = (a_1, a_2, a_3)$

$f(x)_i \neq x$

June 10th 2009 CO759

$e_i = (0, \dots 0, \underset{i}{1}, 0, \dots 0)$

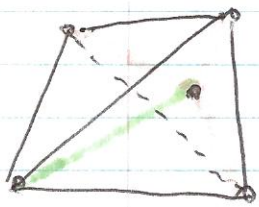
d – dimensional sperner's lemma

Consider the d – dimensional simplex X

$\Delta_{d+1} = \{ x \in \mathbb{R}^{d+1} : \sum_{i=1}^{d+1} x_i = 1, \ x_i \ge 0 \ \forall i \}$

and any point – set $P \subseteq \Delta_{d+1}$ (which includes corner point) and any simplicization of P.

Simplicization of P: connect up pts of P by lines so that every ~~maximal~~ minimal "region" C is convex, all pts. of P in C are extreme pts of C. C has $d+1$ extreme pts.



A valid sperner coloring of P is one where

– the extreme pts $e_1, \dots e_{d+1}$ of Δ_{d+1} get distinct colours $1, 2, \dots d+1$)

– fore every other pt. $x \in P$ x gets the color of one of te extreme pt. $S = \{ e_i : x_i > 0 \}$

Lemma: In any valid sperner coloring of any pt. set $P \subseteq \Delta_{d+1}$, and any simplicization of P. There must exist an odd number of panchromatic small simplexes $\rightarrow$ i.e. a small simplex T where $d+1$ extreme pts get all distinct colours.

proof sketch: Induction on d.

Base cases: $d=1$ $d=2$, — last time

general $d+1$: Create a node for every small simplex and for exterior create an edge b/w simplexes

~~Assume~~ T_1, T_2 if

— $T_1 = \text{conv}(S \cup \{x\})$ $T_2 = \text{conv}(S \cup \{y\})$

— extreme pts of S have all the colours $1, 2, \dots d$ appearing in them.

In this graph,

— external vertex has odd degree. Induction hypothesis

— every vertex has degree ≤ 2

— T is panchromatic simplex $\Rightarrow$ degree $= 1$

$\Rightarrow$ odd # of panchromatic small simplexes.

Proof of Brouwer using sperner's lemma d-dimensional case

will prove for "special" case where $D = \Delta_n$

$\Delta_n = \{x \in \mathbb{R}^n, \sum x_i = 1 \quad x_i \geq 0\}$

$f: D \to D$ continuous.

construct an infinite sequence of pts. in D: for every integer $N \geq 1$.

Construct pt. $y_N \in D$ as follows.

— pick a discrete set $G_N \subseteq \Delta_n$ st. $\forall x, y \in G_N$ ~~$\|x-y\| \leq \frac{1}{N}$~~ and a simplication of G_N where $e_1, \dots e_n \in G_N$

— color G_N as follows: $\forall x, y$ joined by edge, $\|x-y\| \leq \frac{1}{N}$.

— color $x \in G_N$ with $i \in \{1, \dots n\}$ where i is the smallest index st. $x_i > (f(x))_i$

This is a valid Sperner coloring of G_N

$\Rightarrow \exists$ panchromatic simplex by sperner's lemma

$T = \text{conv}(x_1, \dots x_n)$ $x_i \in G_N$

— set $y_N \leftarrow \frac{1}{n}(x_1 + x_2 + \dots + x_n) \in \Delta_n$ convexity

$(y_1, y_2 \dots)$ has a convergent subsequence $\{y_{N_i}\}_{i=1}^{\infty}$

by Bolzano – weierstrass since D is bounded, closed)

Let $y_{N_i} \to y$ claim $y = f(y)$

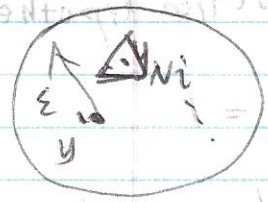
suppose not $\exists\, j$ st. $y_j < f(y)_j$

(1) Since f is continuous, so is $(f(x) - x)_j \Rightarrow \exists$ some small ε

st. $\forall x$ $\|x - y\| \le \varepsilon$, $x_j < f(x)_j$

$\Rightarrow$ No pt in ε-neighbourhood of y is colored j.

(2) But $y_{N_i} \to y \Rightarrow$ can choose i^* st.



$\|y_{N_{i^*}} - y\| \le \varepsilon/3$ and $\frac{1}{N_{i^*}} \le \varepsilon/3 \Rightarrow$

since $y_{N_{i^*}}$ is centroid of some panchromatic simplex $T = \text{conv}(x_1, \dots x_n)$ of $G_{N_{i^*}}$, also here

$\|x_i - y_{N_{i^*}}\| \le \varepsilon/3 \quad \forall i$

But then some x_L in ε-neighbourhood of y is coloured j

$\to$ contradiction $\blacksquare$ QED.

NP-completeness of finding Nash Equilibrium with certain properties

3-SAT

A 3-CNF formula ϕ is with n variables is a conjunction of m clauses $C_1 \wedge C_2 \wedge \dots C_m$ where each

$C_j = x_{j1} \vee x_{j2} \vee x_{j3}$ where each x_{ji} is either x_l or $\overline{x_l}$ $l = 1, 2, \dots n$

[x_l: variable, $x_l, \overline{x_l}$ — literal]

problem is to determine if ϕ has a satisfying assignment.

Given 3-SAT problem formula ϕ, will construct $G(\phi)$, X st. ϕ is satisfied $\Leftrightarrow G(\phi)$ has a NE with total payoff $\ge X$.

will give row-player a strategy set containing $\forall i = 1, \dots n$, strategies (x_i, T) (x_i, F)

Need to ensure (i) Row player "plays" every row x_i with same prob.

(ii) $\forall i$ Row "plays" only one of (x_i, T), (x_i, F)

Building Block I: Matching pennies game

I \ II	H	T
H	(1,-1)	(-1,+1)
T	(+1,+1)	(+1,-1)

Only NE is where both I and II mix their strategies with $p = 1/2$.

$\begin{pmatrix} p > \frac{1}{2} \\ 1-p < \frac{1}{2} \end{pmatrix}$

H	T
$1-2p$	$2p-1$

Take 2 game (R_1, C_1) (R_2, C_2)

R_i, C_i $m \times n$

Row \ col.		
	(R_1+M, C_1+M)	$(1-M, M)$
	$(-M, M)$	(R_2+M, C_2+M)

Assume $M >>$ max entry in R_1, C_1, R_2, C_2

all entries in these two games ≥ 0

June 12th 2008 NP-Completeness proof

Given formula ϕ "other stuff"

construct $G =$

	Q_1	Q_2	$\cdots$	Q_i	Q_n	R
P_1	V_1'	$-M, M$				(X, Y_1)
P_2	$-M, M$	V_2'				(X, Y_2)
$\vdots$			$\ddots$			$\vdots$
P_i	$\vdots$	$\vdots$		V_i'		(X, Y_i)
$\vdots$	$\vdots$	$\vdots$				$\vdots$
P_n	$-M, M$	$-M, M$			V_M'	(X, Y_M)

each off-diagonal block is $(-M, M)$

each diagonal block i is $V_i' = (V_i^{(1)} + M, V_i^{(2)} - M)$

where $V_i = (V_i^{(1)}, V_i^{(2)})$ is a consistency-gadget

every entry of $V_i \geq 0$ and is at most $\alpha M \alpha^{\frac{1}{2}}$ for variable x_i

let $Q = \sum_{i=1}^{n} Q_i$ $\quad X = \begin{pmatrix} 0, 0 \cdots 0 \\ 0, 0 \cdots 0 \\ \varepsilon, \cdots \varepsilon \end{pmatrix}$

Lemma 1:

"Row plays block i w.p. P_i" = total prob. assigned to row of block $i = P_i$ similarly "col plays block i w.p. Q_i".

$P_i\left(1-\frac{\alpha}{2}\frac{n-1}{n}\right] \le \frac{1}{n} \Rightarrow P_i \le \frac{1}{n} + \frac{\alpha}{2n}$ ~~$(n-1)P_i$~~ ~~$\le \frac{1}{n} + \frac{\alpha}{2n}$~~

Lemma 1:

If $Q>0$, then $\forall i$ $|P_i - \frac{1}{n}| \le \frac{\alpha}{2n}$ $|Q_i - \frac{Q}{n}| \le \alpha/2n$

proof: let i be st. $P_i = \max(P_1, \cdots P_n)$ $\alpha \sim \frac{1}{n^2}$

$\cdots$ L be $\cdots$ $P_L = \min(P_1, \cdots P_n)$

will show $\boxed{P_i - P_L \le \frac{\alpha}{2} P_i \ (*)} \Rightarrow P_L \ge P_i(1-\frac{\alpha}{2}) \ge \frac{1}{n} - \frac{\alpha}{2n}$

$P_L \le \frac{1-P_i}{n-1} \Rightarrow P_i \le \frac{1}{n} + \alpha/2n$?

Suppose $(*)$ is false. then col's payoff from playing any strategy in block i $\le M(1-P_i) + P_i(\text{max entry in } V_i^{(2)} - M)$

$\le M(1-2P_i) + \alpha M P_i$

col's payoff from playing any strategy in block L.

$\ge M(1-P_L) + P_L(\text{min entry of } V_L^{(2)} - M) \ge M(1-2P_L)$

If $P_i - P_L > \frac{\alpha}{2} P_i$ then latter payoff $>$ former payoff

$\Rightarrow Q_i = 0 \Rightarrow$ col. puts wp $> \frac{Q}{n}$ on some block $j \ne i$

$\Rightarrow$ Row's payoff from playing a given row in block i

$= -MQ + Z$ — from other stuff

row's payoff from playing corresponding row in block j

$\ge -M(Q-Q_j) + (Q+M)Q_j + Z$ → other stuff

$> -MQ + Z$ (since $Q_j > 0$)

$\Rightarrow$ Row must set $P_i = 0 \Rightarrow$ contradiction

"ANALOGOUS argument for COL".

$V_i = (V_i^{(1)}, V_i^{(2)})$, $V_i' = (V_i^{(1)} + M, V_i^{(2)} - M)$

V_i row \ col	(x_i, T)	(x_i, F)	$*$
(x_i, T)	A, A	B, B	0, A
(x_i, F)	B, B	A, A	0, A
$*$	A 0	A, 0	ε ε

where

$0 < \varepsilon << B < A$

$A < \alpha M$.

Lemma 2: suppose $P_i > 0$, $Q_t > 0$, then within V_i', Row and Col. Both play $\underbrace{\text{either } (x_i, T)}_{\text{exactly one of}}$ or (x_i, F) or $*$.

proof: suppose row mixes and plays (x_i, T) + (x_i, F) wp $P \le P_i$

Then payoff to col look like

(x_i, T)	$x_i F$	$*$
$< Ap$	$< Ap$	$\geq Ap$

$Ap + \Sigma(P_i - P)$

$\Rightarrow$ col plays $*$ exclusively.

$\Rightarrow$ Row would also plays $*$ exclusively. $\rightarrow$ contradiction.

Lemma 3: $Q > 0$ ($\Rightarrow R < 1$)

Lemma 4: $Q = 1$ [$\Rightarrow$ any NE $\equiv$ full or partial assignment] and if assignment is full then it must satisfy $Q\phi$.

By lemma 1~4, every NE of $G = G(\phi)$ is st.

- $|P_i - \frac{1}{n}| \le \frac{\alpha}{2n}$ $Q = 1 \Rightarrow |Q_i - \frac{1}{n} \le \frac{\alpha}{2n}$ $\forall i$.
- By lemma 2, $\Rightarrow$ NE $\equiv$ full or partial assignment.
- If ϕ is unsatisfiable, $\Rightarrow$ every NE is a partial assignment

$\Rightarrow$ gets total payoff $< 2A\sum_i P_i Q_i - (2A - 2\varepsilon)\min P_i Q_i$

if ϕ is satisfiable, then every satisfying assignment yields NE where $P_i = Q_i = 1/n$ $\rightarrow$ total payoff $2A\frac{1}{n}$

Also ensure

June 17th 2008

Lemma 3. $Q > 0$

lemma 4: as above

$C_1 \wedge C_2 \wedge \cdots C_M$

$$X = \begin{array}{c} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{array}{l} \rightarrow (x_i, T) \\ \rightarrow (x_i, F) \end{array} \\ \begin{matrix} \varepsilon & \varepsilon & \cdots & \varepsilon \\ \uparrow & \uparrow & & \uparrow \\ C_1 & C_2 & \cdots & C_M \end{matrix} \quad \rightarrow * \end{array}$$

$$Y_i := \begin{array}{c} \begin{pmatrix} \boxed{} \\ \varepsilon, \varepsilon, \varepsilon, \varepsilon \end{pmatrix} \rightarrow * \\ \begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow \\ C_1 & C_2 & C_j & C_M \end{matrix} \end{array}$$

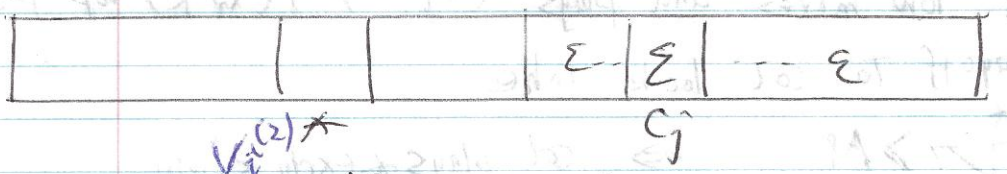
The entry in column C_j and row (x_i, V) is $\begin{cases} -M_1 & \text{if } x_i = V \text{ satisfies } C_j \\ +M_1 & \text{o/w} \end{cases}$

$\forall p \in [\frac{1}{n} - \frac{\alpha}{2n}, \frac{1}{n} + \alpha/2n]$ where $M(1-2p) + Ap < M_1 < M\left(\frac{1-2p}{1-p}\right)$

proof of lemma 3: suppose $Q=0 \Rightarrow R=1 \Rightarrow$

Row will alway play $*$-strategy in every block i w.p. $P_i > 0$

For col. payoff looks like

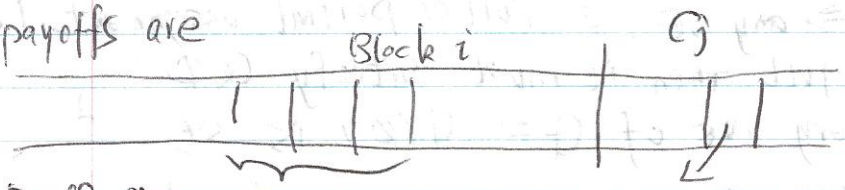


$$(\varepsilon - M)P_i + M(1-P_i) > \varepsilon \Rightarrow R=0$$

Lemma: $Q=1$ 3 cases

case (a) Row (and col) plays a full satisfying assignment.

col's payoffs are



payoff from echoing Row's strategy $AP_i + M(1-2P_i)$

Suppose Row plays (x_i, v) st. C_j is satisfied

payoff $\le M_1(1-P_i) = M_1(1-2P_i)$

$> -M_1 P_i$

$\Rightarrow R=0$

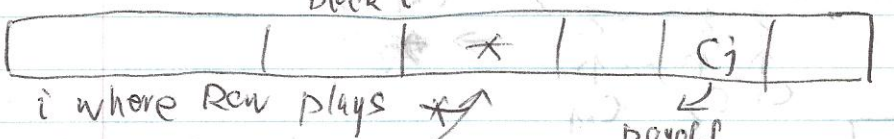
Case (b) Row plays an unsatisfying assignment.

the col's payoffs look similar except $\exists C_j$ that is not satisfied $\Rightarrow$ playing C_j gives payoff $= M_1 >$

payoff from any block $\Rightarrow R=1 \rightarrow$ contradiction.

$M(1-2P_j) + AP_j < M_1$

Case (c) Row plays a partial assignment; col's payoffs are



Z = total prob. wt Row puts on $*$ strategies

i where Row plays $*$

payoff $= (\varepsilon - M)P_i + M(1-P_i)$

payoff $\le \varepsilon Z + (1-Z)M_1$

$= \varepsilon P_i + (1-2P_i)M > \varepsilon Z + (1-Z)M_1$

since $Z > P_i$ and $M_1 < \dfrac{M(1-2P_i)}{1-P_i}$

$\Rightarrow R=0$.

If ϕ is unsatisfiable, then every NE must be a partial assignment with total payoff $\leq 2A\sum P_i Q_i - (2A-2\varepsilon)\min P_j Q_j$

If ϕ is satisfiable with assignment $x_i = v \ \forall i$, then setting $P_i = Q_i = \frac{1}{n}$ and playing (x_i, v) strategy in block i is a NE with payoff $2A\sum P_i Q_i = \frac{2A}{n}$.

verify

Total search problems

A search problem Π is described by

- a set $I_\Pi \subseteq \Sigma^*$ of inputs (Σ^* = set of all strings using symbols of Σ)
- for each $x \in I_n$, a set Δ_n of solutions where each $y \in \Delta_x$ has length polynomial in $|x|$
- a poly time algorithm $B(\cdot,\cdot)$ s.t. $B(x,y) = \text{yes}$ iff $y \in \Delta_x$

A search problem is TOTAL if $\forall x \in I_\pi$, $\Delta_x \neq \phi$.

TFNP = class of all total search PROBLEMS.

Solving a search problem Π means given $B(\cdot)$ and an input $x \in I_\pi$, one seeks to compute $y \in \Delta_x$ (if one exists)

Examples of total search problems

1) r-player NASH an input $x \in I_A$ is an r-player game in normal form, Δ_x = set of all mixed NE of x

2) EQUAL SUBSETS:

An input is n positive integers $a_1, \dots a_n$ s.t. $\sum a_i < 2^n - 1$ a soln is a distinct pair $S \subseteq \{1, \dots n\}$ $T \subseteq \{1, \dots n\}$ s.t.

$$\sum_{i \in S} a_i = \sum_{i \in T} a_i$$

TOTAL by pigeonhole principle

3) END of line: Input directed graph on 2^n nodes with in-out-degree ≤ 1 and source s, soln is a node $v \neq s$

that is a source/sink

Input $x \in I_\pi$ is $(0^n, P=\{0,1\}^n \to \{0,1\}^n, S: \{0,1\}^n \to \{0,1\}^n)$

source

x defines a directed graph $G = (\{0,1\}^n, E = \{(a,b) : b \neq a, S(a)=b, P(b)=a\})$

$a \longrightarrow b$

s.t. $P(0^n) = 0^n$, $S(0^n) \neq 0^n$, $P(S(0^n)) = 0^n$

$[\Rightarrow 0^n$ is a source of $G)$

A soln $y \in \Delta_x$ is a node $b \in \{0,1\}^n$ st. $b \neq 0^n$

and EITHER $P(b)=b$, $S(b) \neq b$ $P(S(b))=b \to$ source

June 19th 2008. OR $S(b)=b$ $P(b) \neq b$, $S(P(b))=b \to$ SINK

Recall

End-of-Line $x \equiv$ directed graph $G = (\{0,1\}^n, E = \{(a,b) : b \neq a \; S(a)=b \; P(b)=a\})$ st. $P(0^n)=0^n$, $S(0^n) \neq 0^n$

soln. y is another source $\neq 0^n$ $\quad 0^n \longrightarrow$ $P(S(0^n)) = 0^n$

or sink of G. denoted as $\Pi_1 \leq \Pi_2$ source of G.

The class PPAD.

Reductions: A polytime reduction from search problem Π_1 to search problem Π_2 consists of a pair of polytime functions f, g such that — $x \in I_{\Pi_1} \Rightarrow f(x) \in I_{\Pi_2}$

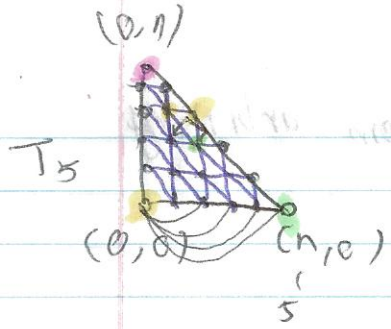
— $y \in \Delta_{f(x)} \Rightarrow g(y) \in \Delta_x$

PPAD = set of all problems Π st $\Pi \leq$ END-OF-LINE.

$= \{\Pi : \Pi \leq$ END of line$\}$

<u>Defn</u>: search problem Π is PPAD-Complete if $\Pi \in$ PPAD & END-OF-Line $\leq \Pi$.

Examples: 1) 2D-SPERNER Will look at triangulations of the form:



$T_n = \{ p = (p_x, p_y) : 0 \le p_x, p_y \le n \;\; p_x + p_y \le n \}$

Input $x \equiv (0^n, A : \{0,1\}^n \times \{0,1\}^n \to \{0,1,2\})$

specifies that triangle is T_{2^n}.

$A(p_x, p_y)$ specifies colour of pt. p.

st. A gives a valid sperner coloring

Soln. $y \in S_x$ is 3 pts $\{p_0, p_1, p_2\}$

st. — $p_0, p_1, p_2 \in T_{2^n}$ and either

$p_1 = p_0 + (0,1) \quad p_2 = p_0 + (1,0)$

OR $p_1 = p_0 - (0,1) \quad p_2 = p_2 - (1,0)$

— trichromatic $\{A(p_0), A(p_1), A(p_3)\} = \{0,1,2\}$

This problem $\in$ PPAD

2) Theorem 1: r-NASH $\in$ PPAD

In fact,

Theorem 2: 2D-sperner is PPAD-complete

Theorem 3: r-NASH is PPAD-complete.

In fact, even 2-NASH is PPAD-complete.

1) r-NASH $\le$ 4-NASH. Goldberg-Papadimitriou

2) Daskalakis-GP: 4-NASH is PPAD-complete

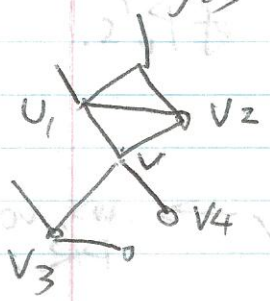
3) DP + Chen-Deng: 3-NASH is PPAD-complete

4) Chen-Deng: 2-NASH is PPAD-complete.

Graphical Games: A graphical game is specified by a graph $G = (V, E)$

— players are nodes of G.

— a player's payoff depends only on its strategy and strategies of its neighbours.



d-graphical game $\equiv$ graph has max. degree $\le d$

d-graphical game where each player has $\le s$ strategies.

$(s=2\ d=3)$

need only $n \cdot s^{d+1}$ numbers $\ll n \cdot s^n$ numbers from arbitrary n-player game.

proof sketch of Theorem 3:

a) 2D-sperner $\leq$ 3-Graphical game

b) 3-graphical game $\leq$ k-NASH (for some constant k)

GP paper

a) produce a graphical game with degree 3 where every player has 2 strategies.

Notation: Each player V with her 2 strategies labeled 0 and 1.
Each player V's mixed strategy can be specified by a number $P_V \in [0,1]$ = prob. player V plays its 1-strategy.

$$0.3125 = 0.25 + 0.0625 = 2^{-2} + 2^{-4} = 0.\underbrace{0101\cdots0}_{n}$$

will have ≥ 3 output nodes, U_0, U_1, V where

P_{U_0}, P_{U_1} specify a pt. $Q^0 = (Q^0_x, Q^0_y)$

specified by P_{U_0} — specified by P_{U_1}

and P_V specifies whether $Q^1 = Q^0 + (1,0)$
$Q^2 = Q^0 + (0,1)$

OR $Q^1 = Q^0 - (1,0)$ $Q^2 = Q^0 - (0,1)$

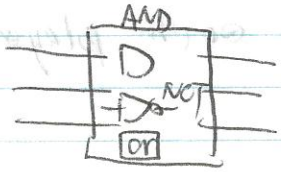
Need to ensure ① Q^0 is a pt. in the interior of T_{2^n}

② Q^0, Q^1, Q^2 gives a trichromatic Δ.

$\Rightarrow$ Need to be able to

- Extract bits from P_V's.
- Arithmetic $\rightarrow$ addition, subtraction, multiplication of P_V's.
- Comparisons $\rightarrow$ $=, <, >$ of P_V's.

eg. 0.3125



circuits

LOGICAL OPERATION $\rightarrow$ AND, NOT, OR.

June 24th, 2008. CO759. Algorithmic Game Theory.

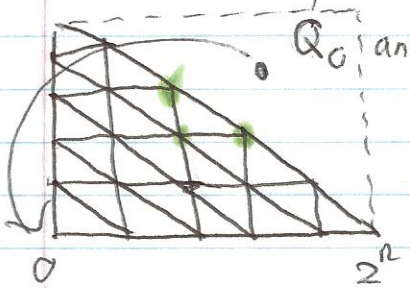
2D-Sperner Input $X \equiv 0^n$, $A: \{0,1\}^n \times \{0,1\}^n \to \{0,1,2\}$ .37.

Solution $\equiv$ trichromatic Δ

Is PPAD-Complete.

2D Sperner $\leq$ 3-Graphical Game

will have 3 output nodes u, v is where P_u, P_v encode a bad vertex



and P_w encodes if $Q_1 = Q_0 + (1,0)$ $Q_2 = Q_0 + (0,1)$

OR $Q_1 = Q_0 - (1,0)$ $Q_2 = Q_0 - (0,$

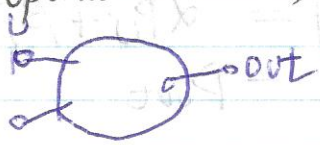
Theorem: There exist graphical-game gadgets that allow us to do

(i) arithmetic (addition, multiplication etc)

(ii) comparisons ($<, >, =$)

(iii) logical operation (AND, OR, NOT)

proof of part (i)



Copy (α): 

Input

Payoffs to w

		out=0	out=1
w:	v=0	0	0
	v=1	α	α

		out=0	out=1
w=1	v=0	0	1
	v=1	0	1

payoffs to out

	w=0	w=1
out=0	0	1
out=1	1	0

If w plays 0, expected payoff $= \alpha P_u$ ($P_u = Pr(u \text{ plays } 1)$)

— — — 1, — — — — — $= P_{out}$

Claim: $P_{out} = \min(1, \alpha P_u)$

If $P_{out} > \alpha P_u$, then w plays 1 w.p. 1 $\Rightarrow$

$P_w = 1 \Rightarrow P_{out} = 0$ — contradiction.

If $P_{out} < \alpha P_u \Rightarrow w$ plays 0 wp. 1 $\Rightarrow$

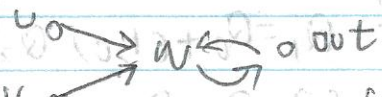
$P_w = 0 \Rightarrow P_{out} = 1 \Rightarrow$

P_{out} cannot be $< \min(1, \alpha P_u)$

$\Rightarrow P_{out} = \min(1, \alpha P_u)$

3 parameters $\alpha, \beta, \gamma \geq 0$

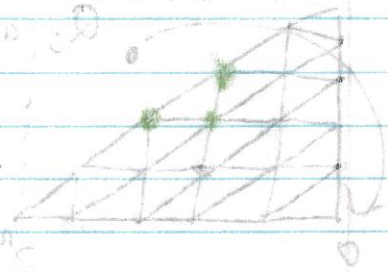
Input



Payoffs to w

$w=0$	$v=0$	$v=1$
$u=0$	0	β
$u=1$	α	$\alpha+\beta+\gamma$

$w=1$	
out 0	0
out 1	1



Payoff to out

	$w=0$	$w=1$
out=0	0	1
out=1	1	0

Claim: $P_{out} = \min(1, \alpha P_u + \beta P_v + \gamma P_u P_v)$

If w plays 0, expected payoff $= \alpha P_u + \beta P_v + \gamma P_u P_v$

--- 1 --- $= P_{out}$

If $P_{out} > \alpha P_u + \beta P_v + \gamma P_u P_v$

$\Rightarrow w$ plays 1 wp. 1 $\Rightarrow P_w = 1 \Rightarrow P_{out} = 0$ ⊠

if $P_{out} < \alpha P_u + \beta P_v + \gamma P_u P_v$

$\Rightarrow w$ plays 0 wp. 1 $\Rightarrow P_w = 0 \Rightarrow P_{out} = 1$

$\Rightarrow P_{out}$ cannot be $< \min(1, \alpha P_u + \beta P_v + \gamma P_u P_v)$

$\Rightarrow P_{out} = \min(1, \alpha P_u + \beta P_v + \gamma P_u P_v)$

Similarly, can construct gadgets for $<, >, =$, AND, OR, NOT.

Notation

out $:= <(a,b) \equiv P_{out} = 1$ if $P_a < P_b$, 0 o/w.

Given these gadgets, and nodes u, v st. $(P_u, P_v) = Q_0$

P_w encoding Q_1, Q_2.

a) Can compute, bit-representation of Q_0, Q_1, Q_2

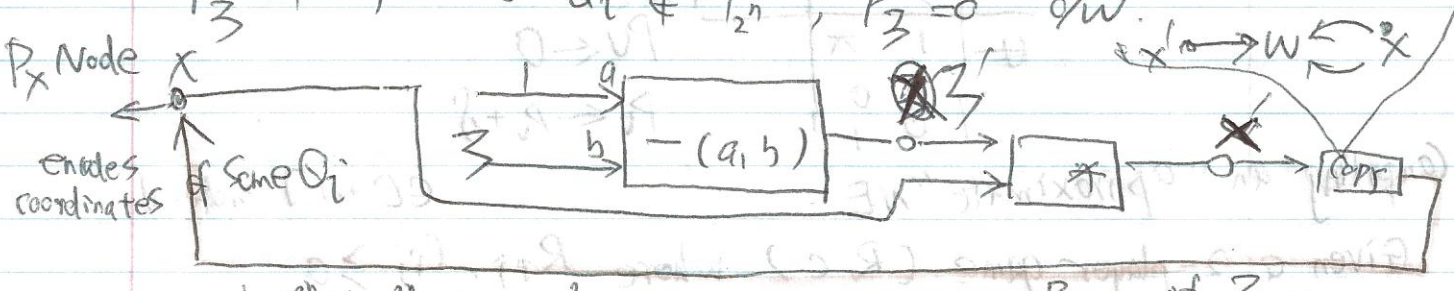
repeat t { If $P_u > a_s 2^{-i} \Rightarrow$ output 1 for 1st bit $b(i) = 1$

$P_u \leftarrow P_u - a_s 2^{-i} \times b(i)$ }

b) Ensure that Q_0, Q_1, Q_2 lie in T_2^n

— first given bit-representations, compute a bit $\mathfrak{Z}$, st. $\mathfrak{Z} = 1$ if some $Q_i \notin T_2^n$ 0 o/w.

I.e, can create graphical gadget with output node $\mathfrak{Z}$ st. $P_{\mathfrak{Z}} = 1$ if some $Q_i \notin T_2^n$, $P_{\mathfrak{Z}} = 0$ o/w.



$\{0,1\}^n \times \{0,1\}^n \to \{0,1,2\}$

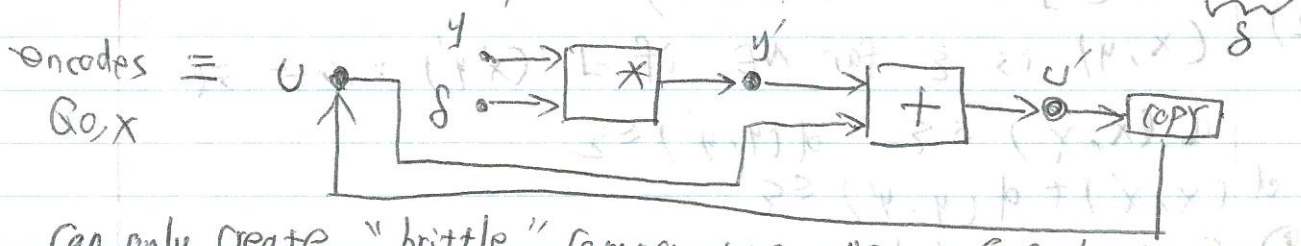
$P_{x'} = \begin{cases} P_x & \text{if } \mathfrak{Z} = 1 \\ 0 & \text{o/w} \end{cases}$

A circuit using AND, OR, NOT.

c) Using bit-representations of Q_0, Q_1, Q_2, and graphical gadgets for AND, OR, NOT can simulate A and compute a bit y st.

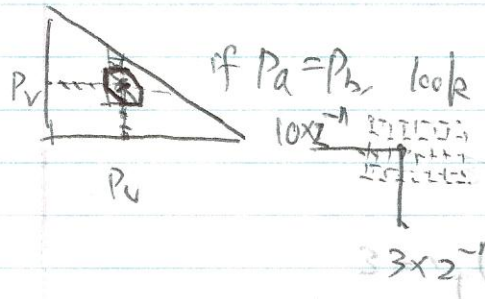
$y = 0$ if Δ is trichromatic
$\quad 1$ o/w

Create a feedback loop that if $y = 1$ sets $Q_0 \leftarrow Q_0 + (\delta, \delta)$



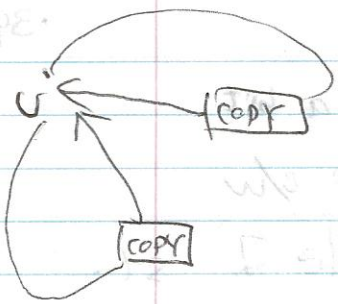
Can only create "brittle" comparators, i.e, a gadget

out := < (a,b) . $P_{out} = 1$ if $P_a < P_b$
$= 0$ if $P_a > P_b$



if $P_a = P_b$, look around

June 26th 2008 CO759 Algorithmic Game Theory



b) Compute a bit $z = 1$ if any of Q_0, Q_1, Q_2 are outside Δ, 0 o/w.

c) Compute a bit $y = 0$ if Δ is trichromatic, 1 o/w.

d)

	z	y	feedback
v	1	*	$P_v \leftarrow 0$
	0	0	$P_v \leftarrow P_v + \delta$
	0	1	

Computing an "approximate" NE. $\in [0,1]$ indeed

Given a 2-player game (R, C) where $R_{ij}, C_{ij} \geq 0$

1) ε-Nash: A mixed strategy profile (x, y) is an ε-Nash Equ if

$\forall i$. $x^T R y \geq e_i^T R y - \varepsilon$ — Additive

$\forall j$, $x^T C y \geq x^T C e_j - \varepsilon$ — multiplicative

$$\left.\begin{array}{l} x^T R y \geq (1-\varepsilon) e_i^T R y \quad \forall i \\ x^T C y \geq (1-\varepsilon) x^T C e_j \quad \forall j \end{array}\right]$$

2) (x, y) is ε-far NE if $\exists$ (x', y') : NE st

$d(x, x') \leq \varepsilon \quad d(y, y') \leq \varepsilon$

$d(x, x') + d(y, y') \leq \varepsilon$

$D((x, y), (x', y')) \leq \varepsilon$.

3) ε-supported NE.

(x, y) is an ε-supported NE if

$\forall i$, $x_i > 0 \Rightarrow (Ry)_i \geq \max_{i'} (Ry)_{i'} - \varepsilon$

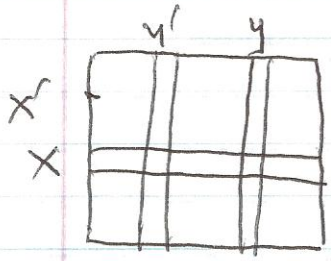
$\forall j$, $y_j > 0 \Rightarrow (x^T C)_j \geq \max_{j'} (x^T C)_{j'} - \varepsilon$

Computing 0.5-NE.

Pick col player, pure strategy y of col.

Let $x \leftarrow$ best response of Row to y.

$y' \leftarrow$ best response of col. to x



Return $(x, 0.5y + 0.5y')$

~~the~~ gain of Row from playing some e_i is $e_i^T R(0.5y + 0.5y') - x^T R(0.5y + 0.5y')$

$= 0.5\underbrace{(e_i^T R y - x^T R y)}_{\leq 0} + 0.5\underbrace{(e_i^T R y - x^T R y')}_{\leq 1} \leq 0$

Gain of Col from playing $e_j = x^T C e_j - x^T C(0.5y + 0.5y')$

$\leq 0.5\underbrace{(x^T C e_j - x^T C y)}_{\leq 1} + 0.5\underbrace{(x^T C e_j - x^T C y')}_{\leq 0} \leq 0.5$

$\exists$ games where any mixed-strategy profile (x, y) where $|\text{support}(x)|, |\text{support}(y)| \leq O(\log N)$

N = total # of strategies of Row, Col. players.

Cannot be better than 0.5-NE.

Computing an improved $\alpha = \frac{3-\sqrt{5}}{2} \approx 0.382$-NE.

Exploits solvability of zero-sum-games.

(2-player) Zero-Sum Game: A game (A, B) where $B = -A$ (so $A + B = 0$)

Can compute a NE efficiently in zero-sum games by solving linear programming (primal-dual LPs)

$$\min \alpha \quad \text{st } \alpha \geq (Ay)_i \quad \forall i \quad \times x_i$$
$$\sum_{j=1}^{n} y_i = 1 \quad \times B$$
$$y_i \geq 0 \quad \forall j$$

$$\max \beta \quad \text{st. } \beta - (x^T A)_j \leq 0 \quad \forall j$$
$$\sum_{i=1}^{n} x_i = 1 \quad \forall i$$
$$x_i \geq 0$$

If (x^*, y^*) is an optimal dual-primal pair then (x^*, y^*) is a NE.

by complementary slackness,

$x^*_i > 0 \Rightarrow \alpha^* = (Ay)_i = \max_{i'} (Ay)_{i'}$

$y^*_j > 0 \Rightarrow -\beta^* = (x^T B)_j = \max_{j'} (x^T B)_{j'}$

Consider zero-sum game $(R-C, C-R)$ let (x^*, y^*) be a NE of this game.

if Row deviates to x', then (*) gain of Col. player (when she continues to play y^*) $\geq$ gain of Row player

$$x'^T C y^* - x^{*T} C y^* \geq x'^T R y^* - x^{*T} R y^*$$

1) compute (x^*, y^*) = NE of $(R-C)(C-R)$

Compute games $g_R = \max_i (Ry^*)_i - x^{*T} R y^*$

$g_C = \max_j (x^{*T} C)_j - x^{*T} C y^*$

2) if $g_R, g_C < \alpha$, then Done

return (x^*, y^*)

3) otherwise $[\max(g_R, g_C) > \alpha]$

a) if $g_R \geq g_C$, then $\begin{cases} s_R \leftarrow \text{best response to } y^* \\ b_C \leftarrow \text{best response to } s_R \end{cases}$

Return $(s_R, (1-\delta_R) y^* + \delta_R b_C)$

where $\delta_R = \frac{1-g_R}{2-g_R}$

b) if $g_C > g_R$ then $\begin{cases} s_C \leftarrow \text{best response to } x^* \\ b_R \leftarrow \text{best response to } s_C \end{cases}$

Return $((1-\delta_C) x^* + \delta_C b_R, s_C)$

where $\delta_C = \frac{1-g_C}{2-g_C}$

(*) $\forall x'$, $x' C y^* - x^{*T} C y^* \geq x'^T R y^* - x^{*T} R y^*$

Analysis: if $g_R, g_C \leq \alpha$ — clearly α-NE

otherwise suppose $g_R \geq g_C$ (other argument is symmetric)

max. gain of Row $\leq \delta_R \cdot 1$ ← verify

max. gain of Col $\leq (1-\delta_R)(\underbrace{\max_j S_R^T C e_j}_{\leq 1} - \underbrace{S_R^T C y^*}_{\geq g_R})$

$g_R = S_R^T R y^* - x^{*T} R y^*$

since taking $x' = S_R$ in $(*)$ $S_R^T R y^* - x^{*T} C y^* \geq S_R^T R y^* - x^{*T} R y^*$

$\Rightarrow$ max gain of Col. $\leq (1-\delta_R)(1-g_R) = \delta_R = \frac{1-g_R}{2-g_R}$

$\Rightarrow$ strategy profile is $\frac{1-g_R}{2-g_R} \leq \frac{1-\alpha}{2-\alpha} = \alpha$ - NE.

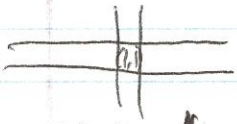
Computing a 0.5 - supported NE for the setting where

$R_{ij}, C_{ij} \in \{0,1\}$

— Compute (x^*, y^*) ← NE of $(R-C, C-R)$

Can assume that $R_{ij} + C_{ij} \leq 1$ $\forall i,j$ o/w.

(no i,j where $R_{ij} = C_{ij} = 1$)



claim: (x^*, y^*) is 0.5- supported NE.

Need to show

$\forall i, i'$, $x_i^* > 0 \Rightarrow (Ry^*)_i \geq (Ry^*)_{i'} - 0.5$

$\forall j, j'$ $y_j^* > 0 \Rightarrow (x^{*T} C)_j \geq (x^{*T} C)_{j'} - 0.5$

$x_i^* > 0 \Rightarrow [(R-C)y^*]_i \geq [(R-C)y^*]_{i'}$

$\Rightarrow (Ry^*)_i - (Ry^*)_{i'} \geq (Cy^*)_i - (Cy^*)_{i'} = C_i \cdot y^* - C_{i'} \cdot y^*$ (dot product)

$\hookrightarrow A_i \equiv i$-th row of A.

claim:

$C_{i'} \cdot y^* - C_i \cdot y^* \leq 1 - (R_{i'} \cdot y^* - R_i \cdot y^*)$

$\equiv \underbrace{(R_{i'} + C_{i'})}_{\text{vector in } \{0,1\}^n} \cdot y^* - \underbrace{(R_i + C_i)}_{\geq 0} \cdot y^* \leq 1$

≤ 1

$\Rightarrow$

$$(Ry^*)_i - (Ry^*)_{i'} \geq -\left((Ry^*)_i - (Ry^*)_{i'} \right) - 1$$

$$\Rightarrow (Ry^*)_i - (Ry^*)_{i'} \geq -0.5$$

July 3rd. CO759

Algorithmic Mechanism Design

Single-item auction: n players, player i has a private value $\bar{V_i}$ for the item

Auctioneer's goal: sell item to player i, st.

$$\bar{V_i} = \max_k \bar{V_k}$$

Auctioneer can charge price P_i to player i (if i wins the item) and this modifies the utility of player i as follows:

$$\text{utility}_i = \begin{cases} \bar{V_i} - P_i & \text{; if } i \text{ wins item.} \\ 0 & \text{; o/w} \end{cases}$$

Auction: allocation rule $f: (b_1, b_2, \cdots b_n) \to$ an outcome $\{e_1, \cdots e_n\}$

pricing scheme $P_i: (b_1, \cdots, b_n) \to \mathbb{R}$ — price charged to player i.

This sets up a game b/w players. $\to$ each player i's strategy is to declare some $b_i \in \mathbb{R}$.

— given a strategy profile $(b_1, \cdots b_n)$

$$\text{utility}(\bar{V_i}; (b_1, \cdots b_n)) = \begin{cases} \bar{V_i} - P_i(b_1, \cdots b_n) & \text{if } i \text{ wins} \\ & \text{i.e. } f(b_1, \cdots b_n) = e_i \\ 0 & \text{o/w} \end{cases}$$

$$\text{i.e. } \bar{V_i}(f(b_i, b_{-i})) - P_i(b_i, b_{-i})$$

where $\bar{V_i}(e_i) = \bar{V_i}$ $\bar{V_i}(e_j) = 0$ $\forall j \neq i$

"stable outcome" of this game: dominant strategy.

Every player i in the above game has a dominant strategy and the dominant strategy is to bid $b_i = \bar{V_i}$

i.e. $\forall i$, $\forall \bar{V_i}$ $\forall b_{-i}$, $\forall b_i$ $\text{utility}_i(\bar{V_i}(\bar{V_i}, b_{-i}))$

$$\geq \text{utility}(\bar{V_i}, (b_i, b_{-i}))$$

"Truthfulness"

Examples of auctions: (Dutch auction)

a) First-price auction.

$f(b_1, \dots b_n) = e_i$ st. $b_i = \max_k b_k$

and $P_i(b) = \begin{cases} b_i & \text{if } e_i = f(b) \\ 0 & \text{o/w} \end{cases}$

— No dominant strategies, not truthful.

b) Fixed-winner auction $f(b) = e_1$

$P_1(b) = t\$$, $P_j(b) = 0$ $\forall j \neq 1$.

— Dominant strategy(ies) exist.

Implement auctioneer's goal.

c) Second-price auction. $f(b) = e_i$ st. $b_i = \max_k b_k$ no collision.

$P_i(b) = \begin{cases} \max_{k \neq i} b_k & \text{if } f(b) = e_i \\ 0 & \text{o/w} \end{cases}$

Fixed player i, let $b^* = \max_{k \neq i} b_k$

1) $\bar{V}_i \geq b^*$. if i bids $\geq b^*$, he wins, gets utility $\bar{V}_i - b^*$ $\Rightarrow$ in particular, bidding $\bar{V}_i$ earns utility $\bar{V}_i - b^*$.

2) $\bar{V}_i < b^*$: max utility i can get $= 0$, and he earns this utility by bidding $\bar{V}_i$.

Truthful auction and hence implements the auctioneer's goal of allotting item to player i with $\bar{V}_i = \max_k \bar{V}_k$

Definitions and Notations:

— A: set of alternatives / outcomes = set of all outputs to the problem

— n players

— each player i has a true private valuation ~~of~~function $\bar{V}_i : A \to \mathbb{R}$. $\bar{V}_i(a)$ = value i assigns to outcome $a \in A$

— $V_i \subseteq \mathbb{R}^A$ = set of all allowed valuation functions of player i

Notation: $V = V_1 \times \cdots \times V_n$ $\quad \underline{V}_i = \prod_{j \neq i} V_j$

Defn: A mechanism M is a tuple $(f, p = \{p_i\})$ where

- $f : V \to A$ $\equiv$ algorithm / allocation rule that maps every input valuation vector $v \in V$ to an outcome $a \in A$

  Also called social-choice ~~rule~~ function.

- $p_i : V \to \mathbb{R}$ $\equiv$ $p_i(v)$ price i pays when the declared valuation vector of players is $v \in V$.

Given mechanism M, the utility of i under a strategy profile $v \in V$ is $\text{utility}_i(\bar{v}_i ; (v_i, v_{-i})) = \bar{v}_i(f(v)) - p_i(v)$

eg. single-item auction, $A = \{e_1, \ldots e_n\}$

$\bar{v}_i(a) = \bar{v}_i e_i^T a \quad \forall a \in A$. $e_i \equiv$ vector with 1 in i-th position

Defn. A mechanism $M = (f, p)$ is truthful if

$\forall i, \forall \bar{v}_i, \forall b_{-i} \in \underline{V}_i, \forall b_i \in V_i$

$\text{utility}_i(\bar{v}_i, (\bar{v}_i, b_{-i})) \geq \text{utility}(\bar{v}_i, (b_i, b_{-i}))$

$\equiv$ $\text{utility}_i(\bar{v}_i, (\cdot, b_{-i}))$ is maximized at $\bar{v}_i$.

Also called dominant-strategy incentive compatibility.

BASIC Question in AMD: Given a "target" SCF (social choice function) g ~~VA~~. does there exist a pricing scheme, i.e. $p_i : V \to \mathbb{R}$ $\forall i = 1, \ldots n$ st.

$M = (g, \{p_i\})$ is truthful.

If $\exists p_i$ we say that g is DS-implementable or simply g is truthfully implementable or g is implementable.

AMD: Is g implementable efficiently.

Question: Which social choice function g are (efficiently) implementable?

July 8th 2008 CO759 Algorithmic game theory.

n players.

A = set of alternatives each i has a set $V_i \subseteq \mathbb{R}^A$

set of all possible input of i

Given a target function $g: V \to A$, can one (efficiently) implement g, i.e. $(V = V_1 \times \cdots \times V_n)$

do $\exists P_i: V \to \mathbb{R}$ $\forall i = 1, \cdots n$, s.t. $M = (g, \{P_i\})$ is a truthful mechanism?

often $g: V \to A$ has the form $g(v) = a^*$ where a^* is an optimal soln. to some optimization problem, i.e, a^* optimizes (max/min) $\pi(v, a)$ over all $a \in A$

eg. Single-item auction: $g(v) = a^*$ maximizes $\sum V_i(a)$ over all $a \in A$

eg. MST problem: Given a graph G. → common knowledge. Each edge e is a player whose private value is the cost C_e of edge e. $A = \{$all spanning trees of $G\}$

Each player e has a valuation function $\bar{C}_e(T) = \begin{cases} -\bar{C}_e; & \text{if } e \in T \\ 0 & \text{o/w} \end{cases}$

$V_e = \{$all functions of the above form$\}$

$g: V \to A$ where $g(c)$ = MST wrt. edge costs C_e $= T^*$ that maximizes $\sum_e C_e(T)$ over all $T \in A$

If g is a soln to an NP-hard problem, cannot expect to be able to compute g efficiently. In this case, $\exists$ some function $f: V \to A$ st.

(1) f is an α-approximation algorithm for π. —

— f can be computed efficiently.

— $\forall v \in V$ $\pi(v, f(v))$ is an α-factor away from $\pi(v, g(v))$

(2) f is efficiently implementable.

Truthful Mechanism Design.

Vickery-Clarke-Groves (VCG) Mechanism (Family)

Consider $g: V \to A$. where $g(v) = a^*$ that maximizes

$\underbrace{\sum_i v_i(a)}_{\text{social welfare objective.}}$ over all $a \in A$.

$g(v) =$ opt. soln. to π where $\pi(v, a) = \sum_i v_i(a)$

social welfare maximization problem.

Fix i, fix V_{-i}. let $a = g(\bar{v}_i, v_{-i})$

$\bar{v}_i \in V_i \subseteq \mathbb{R}^A$ is $\qquad b = g(v_i, v_{-i}) \quad v_i \in V_i$

i's true private valuation $\Rightarrow$ (1) $\bar{v}_i(a) + \sum_{j \neq i} v_j(a) \geq v_i(b) + \sum_{j \neq i} v_j(b)$

follows from defn. of g.

Truthfulness means: (2) $\bar{v}_i(a) - P_i(\bar{v}_i, v_{-i}) \geq v_i(b) - P_i(v_i, v_{-i})$

Can ensure (2) by setting $P_i(\bar{v}_i, v_{-i}) = -\sum_{j \neq i} v_j(a)$

$P_i(v_i, v_{-i}) = -\sum_{j \neq i} v_j(b)$

Can set $\boxed{P_i(v) = -\sum_{j \neq i} v_j(g(v))}$

setting $P_i(v) = -\sum_{j \neq i} v_j(g(v))$ ensures $M = (g, \{P_i\})$ is truthful.

Easy fix to above prices: $P_i(v) = -\sum_{j \neq i} v_j(g(v)) + h_i(v_{-i})$

Theorem: (VCG) In any mechanism-design setup, i.e. $A, V_1, \dots V_n \subseteq \mathbb{R}^A$ then the SCF $g: V \to A$ where $g(v) = a^*$ that maximizes $\sum_i v_i(a)$ over all $a \in A$ is implementable using any prices $P_i: V \to \mathbb{R}$ of the form $P_i(v) = -\sum_{j \neq i} v_j(g(v)) + h_i(v_{-i})$

Examples:

1) Single-item auction: $h_i(v_{-i}) = \sum_{j \neq i} v_j(b)$ where $b \in A$ maximizes social welfare where i is <u>NOT</u> a player.
gives 2nd price auction rule.

2) Shortest s-t path. graph $G=(V,E)$ s,t -
$A = \{ P : P$ is an s-t path in $G \}$
every edge is a player, with valuation $\bar{c}_e(P) = \begin{cases} -\bar{c}_e & \text{; if } e \in P \\ 0 & \text{; o/w.} \end{cases}$
$V_e = \{$all function $\bar{c}_e$ of above form$\}$
$g : V \to A$ where $g(c) = p^*$ that maximizes $\sum_e c_e(P)$ over $P \in A$.
$\Rightarrow$ can use VCG, and set $P_e(c) = -\sum_{e' \neq e} c_{e'}(P) + h_e(c_{-e})$
want to ensure a player has non-negative utility if he plays truthfully. (individual rationality or voluntary participation)

$$\text{utility}_e(\bar{c}_e ; (\bar{c}_e, c_{-e})) = \underbrace{\bar{c}_e(P) + \sum_{e' \neq e} c_{e'}(P)}_{= -\text{ve cost of } P} - h_e(c_{-e})$$

Assume s, t are 2-edge connected,
set $h_e(c_{-e}) = -$ cost of shortest s-t path in $G \setminus \{e\}$.
$= -c_e(P_e) \equiv$ max SW when e is not in the game
Then $\text{utility}_e(\bar{c}_e ; (\bar{c}_e, c_{-e})) = \text{cost}(P_e) - \text{cost}(P) \geq 0$
These h_e's also ensure.

(1) $P_e(\bar{c}_e, c_{-e}) \leq 0$

(2) $P_e(\bar{c}_e, c_{-e}) = 0$ if $e \notin g(\bar{c}_e, c_{-e})$

no positive transfers.

Limitations of VCG.

1) What if computing g is intractable?
Then, want to implement an α-approx. $\hat{g}$ to g.

One option: set prices as given by VCG pricing scheme, i.e.

$$P_i(v) = -\sum_{j \neq i} v_j(\hat{g}(v)) + \cancel{h_i(v_{-i})}$$

July 10th 2008 CO759 Algorithmic Game Theory

Limitations of VCG.

1) Suppose computing $g: V \to A$ where $g(v) = \arg\max_{a \in A} \sum_i v_i(a)$ is $\mathcal{NP}$-hard, then cannot simply take some approx. algorithm $\hat{g}$ for the problem and use this with VCG prices ($P_i(v) = -\sum_{j \neq i} v_j(\hat{g}(v)) + h_i(v_{-i})$) and get a truthful mechanism.

2) VCG only works for social-welfare maximization (SWM) problems, what about other objectives?

Single-dimensional domains: Restrict V_i's and give a complete "algorithmically convenient" characterization of which SCFs $g: V \to A$ are implementable.

A: alternative set, n players.

Defn: A domain $V_i \subseteq \mathbb{R}^A$ is called single dimensional if every $v_i \in V_i$ has the form $v_i(a) = v_i \alpha_i(a)$ where $\alpha_i \in \mathbb{R}^A$ is a fixed vector that is common knowledge.

$V_i = \{ v_i \alpha_i(a) : v_i \in N_i \subseteq \mathbb{R} \}$

Examples: (1) single-item auction: $\alpha_i(a) = \begin{cases} 1 & \text{if } a = e_i \\ 0 & \text{o/w} \end{cases}$

(2) MST problem: $V_e(T) = -c_e \alpha_e(T)$ $\alpha_e(T) = \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{o/w} \end{cases}$

$V_e = \{ v_e = -c_e v_e : c_e \geq 0 \}$

Notation: Frequently, will have $\alpha_i \in \{0,1\}^A$

$W_i = \{ a \in A : \alpha_i(a) = 1 \}$ $L_i = \{ a \in A : \alpha_i(a) = 0 \}$

Can view an alternative a as a $\{0,1\}^n$ where $a_i = \begin{cases} 1 & \text{if } a \in W_i \\ 0 & \text{if } a \in L_i \end{cases}$ $a_i = a_i(a)$

(3) Max. indept. set problem:

$G \to$ common knowledge.

each vertex x is a player, $A = \{ I \subseteq N : I$ is an ind. set of $G \}$.

$\alpha_x(I) = \begin{cases} 1 & \text{if } x \in I \\ 0 & \text{o/w} \end{cases}$ $\quad V_x(I) = V_x \cdot \alpha_x(I)$ where $V_x \geq 0$.

Suppose each player i owns a set $S_i \subseteq N$ of nodes and its private information is $(V_x)_{x \in S_i}$, $V_x \geq 0 \ \forall x$.

$$V_i(I) = \sum_{x \in S_i \cap I} V_x$$ — NOT SINGLE DIMENSIONAL.

Becomes single dimensional if $V_x = V_i \ \forall x \in S_i$

$V_i(I) = V_i |I \cap S_i| = V_i \alpha_i(I)$ where

$\alpha_i(I) = |I \cap S_i|$

Theorem: ~~Let~~ suppose all V_is are single dimensional.

Then $g: V \to A$ is implementable iff $\forall i, \forall V_{-i} \in V_{-i}$.

$\alpha_i(g(V, V_{-i}))$ is an increasing func. of V.

monotonicity property.

sanity check: For SP-problem, $g: V \to A$ which outputs the SP satisfies Theorem.

single dim $\to$ $V_i \subseteq \mathbb{R}^A$ is of the form $\{ V_i \alpha_i : V_i \in N \subseteq \mathbb{R} \}$

Lemma 1: [necessary condition for implementability in <u>any domain</u> (not just single-dimensional)].

Let $A, V_1, \cdots V_r$ be a mechanism-design domain.

Then $g: V \to A$ is implementable only if $\forall i, \forall V_{-i} \in V_{-i}$

$\forall V_i, V_i' \in V_i$, letting $a = g(V_i, V_{-i})$ $b = g(V_i', V_{-i})$

We have $v_i(a) + v_i'(b) \geq v_i(b) + v_i'(a)$

[equivalently, $v_i(a) - v_i(b) \geq v_i'(a) - v_i'(b)$] $(*)$

Proof: g is implementable $\Rightarrow \exists P_j : V_j \to \mathbb{R}$ st.

$M = (g, \{P_j\})$ is a truthful Mechanism.

Fix i, $V_{-i} \in V_{-i}$.

Consider $v_i, v_i' \in V_i$ let $a = g(v_i, v_{-i})$ $b = g(v_i', v_{-i})$

TRUTHFULNESS $\Rightarrow$ $\text{utility}_i(v_i; (v_i, v_{-i})) \geq \text{utility}_i(v_i; (v_i', v_{-i}))$

$\equiv$ $v_i(a) - P_i(v_i, v_{-i}) \geq v_i(b) - P_i(v_i', v_{-i})$ ....(1)

$\text{utility}(v_i', (v_i' v_{-i}) \geq \text{utility}(v_i', (v_i, v_{-i}))$

$\equiv$ $v_i'(b) - P_i(v_i', v_{-i}) \geq v_i'(a) - P_i(v_i, v_{-i})$ (2)

(1) + (2) gives statement

Lemma 2: [specialization of Lemma 1 for Single-Dim. Domains]

If all V_is are SD, $g: V \to A$ is implementable only if g satisfies monotonicity property, i.e. $\alpha_i(g(v, v_{-i}))$ is an $\uparrow$ func. of v $\forall i$, $\forall v_{-i}$

proof: Fix i, v_{-i}, let $v_i, v_i' \in V_i$. st. $v_i \geq v_i'$

$\cancel{v_i(\bar{a}) = v_i\ \alpha_i(\bar{a})}$

$v_i(\cdot) = v_i\ \alpha_i(\cdot)$

$v_i'(\cdot) = v_i'\ \alpha_i(\cdot)$

Let $a = g(v_i, v_{-i})$ $b = g(v_i', v_{-i})$

To show: $\alpha_i(a) \geq \alpha_i(b)$

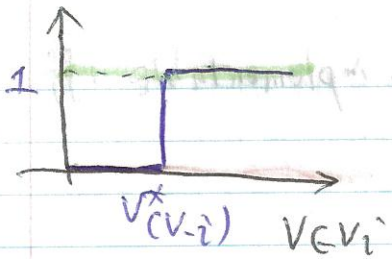
We have: from $(*)$

$v_i[\alpha_i(a) - \alpha_i(b)] \geq v_i'[\alpha_i(a) - \alpha_i(b)]$

$\equiv$ $(v_i - v_i')(\alpha_i(a) - \alpha_i(b)) \geq 0 \Rightarrow$

$\alpha_i(a) \geq \alpha_i(b)$ since $v_i \geq v_i'$

$\alpha_i(g(v, v_{-i}))$ Fixing i. v_{-i}.



Consider $\alpha_i \in \{0, 1\}^A$

$v_i(a) = v_i \, \alpha_i(a) \quad v_i \geq 0$

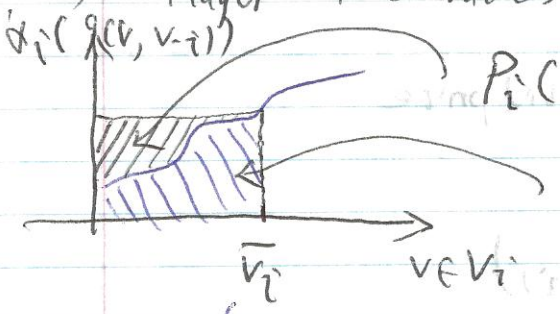
Lemma 3: Suppose all V_is are single-dimensional. Then g is implementable, if g satisfies monotonicity property.

The condition $\forall i, \forall v_{-i}, \forall v_i, v_i' \in V_i$, letting $a = g(v_i, v_{-i})$, $b = g(v_i', v_{-i})$ $\quad v_i(a) + v_i'(b) \geq v_i(b) + v_i'(a)$

which is necessary for implementability is called <u>Weak MONOTONICITY</u>

Proof: Fix i, v_{-i}. Break up into 2 settings.

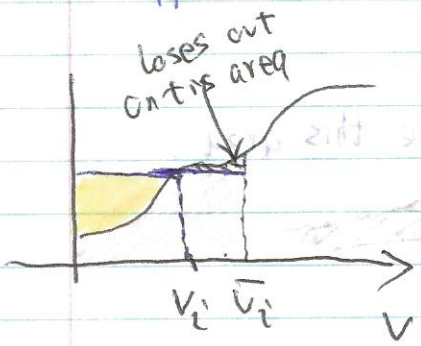
1) "Players have values": $v_i(a) = v_i \, \alpha_i(a)$, $v_i \geq 0$



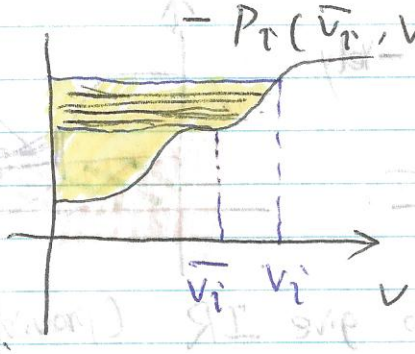
$$P_i(\bar{v}_i, v_{-i}) = \int_0^{\bar{v}_i} [\alpha_i(g(\bar{v}_i, v_{-i})) - \alpha_i(g(x, v_{-i}))] \, dx$$

$$\text{Utility}_i(\bar{v}_i, (\bar{v}_i, v_{-i})) = \bar{v}_i \cdot \alpha_i(g(\bar{v}_i, v_{-i}))$$

Suppose true value of i.



$- P_i(\bar{v}_i, v_{-i})$



$$\Rightarrow P_i(v_i, v_{-i}) = \int_0^{v_i} [\alpha_i(g(v_i, v_{-i})) - \alpha_i(g(x, v_{-i}))] \, dx$$

implement g.

CS 759 July 17th 2008

Theorem: Let $A, V_1, \ldots V_n$ be a mechanism-design setting with

all v_is single dimensional. Then $g: V \to A$ is implementable iff

$\forall i, \forall v_{-i}$ $\alpha_i(g(v, v_{-i}))$ is $\nearrow$ in v.

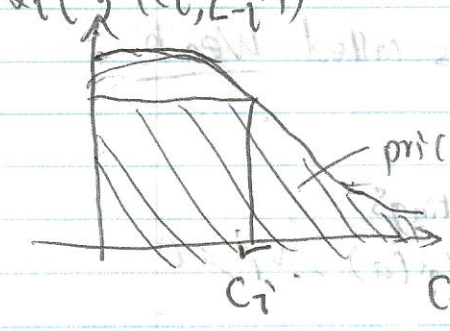
proof: LAST TIME. Necessity.

— proved sufficiency for "value" players:

$$P_i(v_i, v_{-i}) = \int_0^{v_i} [\alpha_i(g(v_i, v_{-i})) - \alpha_i(g(x, v_{-i}))]\,dx$$

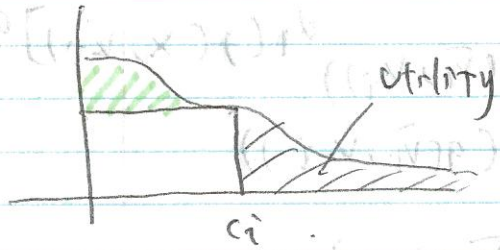
VERIFY that these prices also yield individual rationality. (IR)

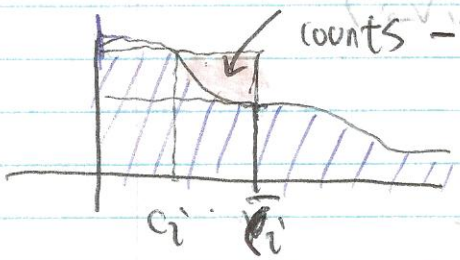
Now consider "cost" players: $v_i(a) = -c_i \alpha_i(a)$ where $c_i \geq 0$

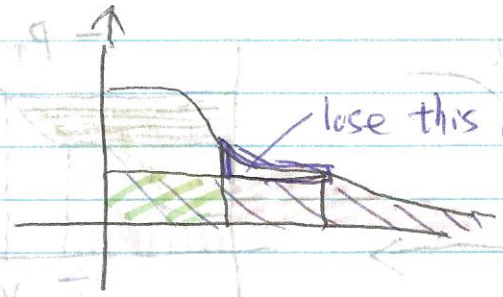


Define $P_i(\bar{c}_i, c_{-i}) = \Big[\bar{c}_i \cdot \alpha_i(g(\bar{c}_i, c_{-i})) + \int_{\bar{c}_i}^{+\infty} \alpha_i(g(x, c_{-i}))\,dx\Big]$

$\bar{c}_i$ bid price







Verify these prices also give IR. (provided

$$\int_{c_i}^{\infty} \alpha_i(g(x, c_{-i}))\,dx < \infty)$$

In general any prices of the form $P_i(c_i, c_{-i})$

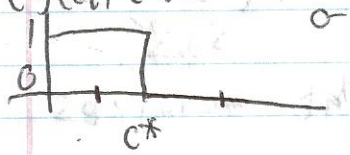
$$= \int_0^{c_i} [\alpha_i(g(x, c_{-i}) - \alpha_i(g(c_i, c_{-i}))]\,dx$$

$+ h_i(c_{-i})$ gives truthfulness.

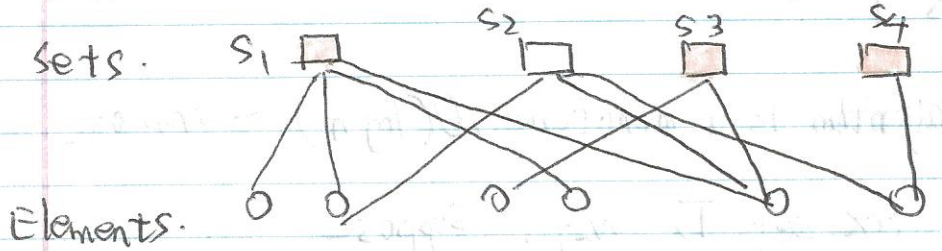
VERIFY. that for MST s-t SP problem, single-item auction, prices computed by Theorem coincide with VCG prices.

Applications of Monotonicity Characterization.

$x_i(g(c_i, c_{-i}))$ 0-1 function



1) SET COVER: Given a universe X of m elements, and a collection $\mathcal{S}$ of m subsets of X. Each set $S \in \mathcal{S}$ has a weight/cost W_S GOAL: Compute a min-wt. collection $\mathcal{S}' \subseteq \mathcal{S}$ st. every element $e \in X$ is contained in some set $S \in \mathcal{S}'$ $\mathcal{S}'$ is called a set cover.



In mechanism design setting, every set S is a player whose private value is $W_S \geq 0$

$A = \{\mathcal{S}' \subseteq \mathcal{S} : \mathcal{S}'$ is a set cover$\}$

Each player S's domain is $V_S = \{v_S \in \mathbb{R}^A : v_S = -W_S \alpha_S \quad W_S \geq 0$

where $\alpha_S(\mathcal{S}') = \begin{cases} 1 & \text{if } S \in \mathcal{S}' \\ 0 & \text{o/w} \end{cases}$

Goal: Devise an approx. algorithm $g: V \to A$ st. g is implementable

i.e. g satisfies monotonicity property.

Monotonicity $\equiv \forall S, \forall W_{-S}$: $\alpha_S(g(W, W_{-S}))$ ↓ of/with W.

$\equiv$ If $S \in \mathcal{S}' = g(W, W_{-S})$, then for $W' < W$. $S \in \mathcal{S}'' = g(W', W_{-S})$.

Approximation algorithm: "Greedy Algorithm"

1) Initialize $\mathcal{S}' \leftarrow \emptyset$ $\quad X' \leftarrow X$.

[X' is a set of uncovered elements given that sets in $\mathcal{S}'$ have been picked, ie. $X' = X \setminus \bigcup_{S \in \mathcal{S}'} S$]

2) while $X' \neq \emptyset$ – chose $T \in \mathcal{S} \setminus \mathcal{S}'$ that minimizes

$$\frac{w_S}{|S \cap X'|} \text{ over all } S \in \mathcal{S} \setminus \mathcal{S}'$$

– $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{T\}$ $\quad X' \leftarrow X' \setminus T$.

3) Return $\mathcal{S}'$.

[Assume $\bigcup_{S \in \mathcal{S}} S = X$].

Theorem: The greedy algorithm is a monotone, $O(\log n)$ – approx. algorithm.

proof: Monotone: Fix set T, w_{-T}. Suppose

$T \in \mathcal{S}' = g(w, w_{-T})$,

Let $S_1, S_2, \dots \underset{S_i}{T} - S_{i+1}, \dots S_k$ be the sets picked in order

let $\mathcal{S}_i' = \{S_1, \dots S_i\}$

Let $X_i = X'$ just before set S_i is picked.

Greedy $\Rightarrow \forall i$, $\dfrac{w_{S_i}}{|S_i \cap X_i|} = \min_{S \notin \mathcal{S}_{i-1}'} \dfrac{w_S}{|S \cap X_i|}$

Consider input (w', w_{-T}) $w' < w$.

Let i be smallest index st.

$$\frac{w'}{|T \cap X_i|} = \min_{S \notin \mathcal{S}_{i-1}} \frac{w_S}{|S \cap X_i|}$$

[Note that $i \leq j$, since $\dfrac{w}{|T \cap X_j|} = \min_{S \notin \mathcal{S}_{j-1}} \dfrac{w_S}{|S \cap X_j|}$

$\rightarrow$ In iterations $1, \dots i-1$, [on input (w', w_{-T})], pick same $S_1, \dots S_{i-1}$, greed-choice rule $\Rightarrow$

$T \in \mathcal{S}'' = g(w', w_{-T})$

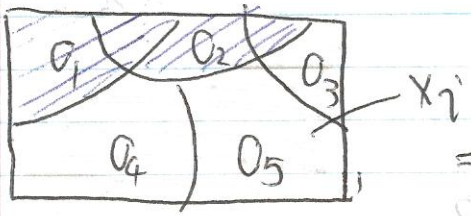
Moreover, note that threshold u^* at which $x_T(g(\cdot\ w_{-T}))$ goes from $1 \to 0$ can be computed efficiently.

Approximation. Let $n_i = |X_i|$ let $\{O_1, O_2, \dots O_L\}$ optimal soln. $= \mathcal{O}$

Claim: $\dfrac{w_{S_i}}{|S_i \cap X_i|} \le \dfrac{OPT}{n_i}$ $\qquad OPT = w_{O_1} + w_{O_2} + \dots w_{O_L}$

proof: Consider the sets in $\mathcal{O} \setminus \mathcal{S}_{i-1}$,

$\sum_{S \in \mathcal{O} \setminus \mathcal{S}_{i-1}} w_S \le OPT$, $\quad \bigcup_{S \in \mathcal{O} \setminus \mathcal{S}_{i-1}} (S \cap X_i) = X_i$



$\Rightarrow \sum_{S \in \mathcal{O} \setminus \mathcal{S}_{i-1}} |S \cap X_i| \ge n_i$

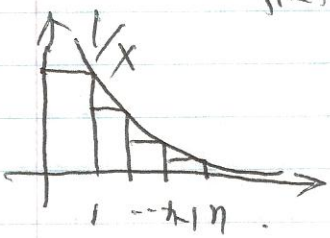
$\Rightarrow \exists$ some set $S \in \mathcal{O} \setminus \mathcal{S}_{i-1}$ st

$$\frac{w_S}{|S \cap X_i|} \le \frac{OPT}{n_i}$$

$$\Rightarrow \frac{w_{S_i}}{|S_i \cap X_i|} \le \frac{OPT}{n_i} \qquad \sum_{S \in \mathcal{S}'} w_S = \sum_{i=1}^{k} w(S_i)$$

$$= \sum_{i=1}^{k} \frac{w(S_i)}{|S \cap X_i|} \cdot |S_i \cap X_i| \le \sum_{i=1}^{k} \frac{OPT}{n_i} \cdot (n_i - n_{i+1})$$

$$\le OPT\left(\sum_{i=1}^{k} \sum_{x=n_{i+1}+1}^{n_i} \frac{1}{x}\right) = OPT \cdot \underbrace{\sum_{x=1}^{n} \frac{1}{x}}_{H_n} \le OPT \cdot (1 + \log n)$$



July 22nd CO759 Algorithmic Game Theory 2008.

Facility Location (METRIC): A set F of m facilities
a set C of n clients.
Each facility i has an opening cost $f_i \geq 0$
Assume facilities are located in a common metric space
and for each client $j \in C$, facility $i \in F$, there is a
cost $c_{ij} \geq 0$ of assigning j to i = distance b/w i and j
Goal: choose a set $X \subseteq F$ of facilities to open and assign
each $j \in C$ to an open facility $i(j) \in X$ to

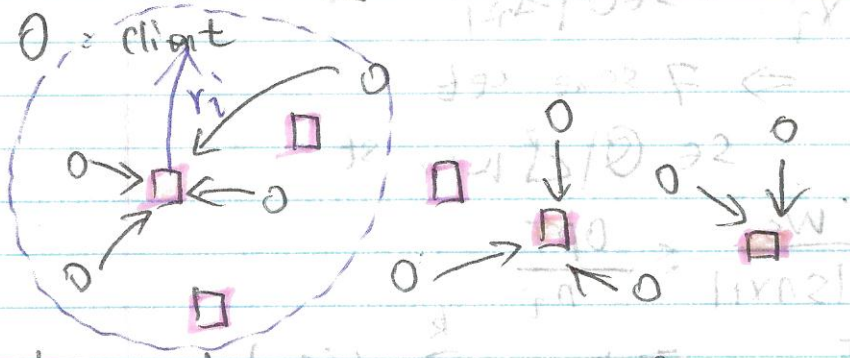
minimize $\sum_{i \in X} f_i + \sum_{j \in C} c_{i(j)j}$

Given set $X \subseteq F$, $X \neq \emptyset$, will assign each j to nearest facility
in X $\Rightarrow$ cost of soln is

$\text{cost}(X) := \sum_{i \in X} f_i + \sum_{j \in C} c(j, X)$ where $c(j, X) = \min_{i \in X} c_{ij}$

□ : facility ▣ : open facility

O : client



Mechanism design setting: Each facility $i \in F$ is a player
where private information is his opening cost $f_i \geq 0$
$A = \{X \subseteq F : X \neq \emptyset\}$ each player i has a single
dimensional domain $V_i = \{v_i \in \mathbb{R}^A : v_i = -f_i \alpha_i\}$
where $\alpha_i(X) = \begin{cases} 1, & \text{if } i \in X \\ 0 & : \text{o/w.} \end{cases}$

Monotonicity $\equiv$ if $i \in X = g(f, f_{-i})$
Then $i \in X' = g(f', f_{-i})$ for $f' < f$

Notation: $C(j,X) = \min_{i \in X} c_{ij}$ $\quad$ $\text{cost}(X) = \sum_{i \in X} f_i + \sum_{j \in C} c(j,X)$

Algorithm: for some $x \in F \cup C$, $r \geq 0$ define

$$B(x,r) = \{ y \in F \cup C : c_{xy} \leq r \}$$

for each $i \in F$, define r_i (radius of i) to be smallest $r \geq 0$ st.

$$\sum_{j \in C \cap B(i,r)} (r - c_{ij}) = f_i$$

Note: r_i is an $\nearrow$ func. of f_i.

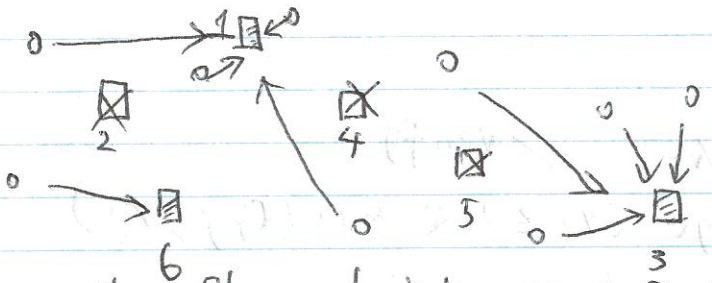
Algorithm: 1) Sort the facilities so that $r_1 \leq r_2 \leq \dots \leq r_m$

2) Initialize $Z \leftarrow \emptyset$.

3) For $i = 1, \dots, m$, if $c_{ii'} > 2r_i$, $\forall i' \in Z$, set

$Z \leftarrow Z \cup \{i\}$

4) Return Z.



Theorem: The above algorithm is a 3-approx. algorithm that is implementable.

proof: Monotonicity — Exercise. (follows easily from monotonicity of r_i's)

Approximation: Define Given a soln $X \subseteq F$, $X \neq \emptyset$,

client j, $\text{Charge}(j,X) = c(j,X) + \sum_{i \in X} \max(0, r_i - c_{ij})$

Lemma 1: $\forall X \neq \emptyset$, $\sum_{j \in C} \text{Charge}(j,X) = \text{cost}(X)$

proof: $\sum_{j \in C} \text{charge}(j,X) = \sum_{j \in C} c(j,X) + \underbrace{\sum_{i \in X} \sum_{j \in C} \max(0, r_i - c_{ij})}$

$$= \sum_{i \in X} \sum_{j \in C \cap B(i,r_i)} (r_i - c_{ij}) = \sum_{i \in X} f_i$$

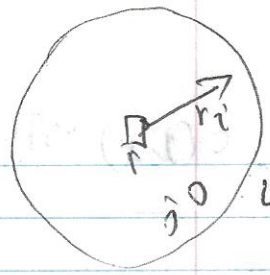
Goal: $\forall X \neq \emptyset, \subseteq F$,

$\forall j$, $\text{Charge}(j,Z) \leq 3 \cdot \text{charge}(j,X)$

Fact 2: If $i, i' \in Z$, then $c_{ii'} > 2\max(r_i, r_{i'})$

Hence $B(i, r_i) \cap B(i', r_{i'}) = \emptyset$ and if

$j \in C \cap B(i, r_i)$ then



$c(j, Z) = c_{ij}$ and $\text{charge}(j, Z) = r_i$

Lemma 3: For any $i \in F$, then $\exists i' \in Z$ st. $i' \leq i$ and $c_{ii'} \leq 2r_i$.

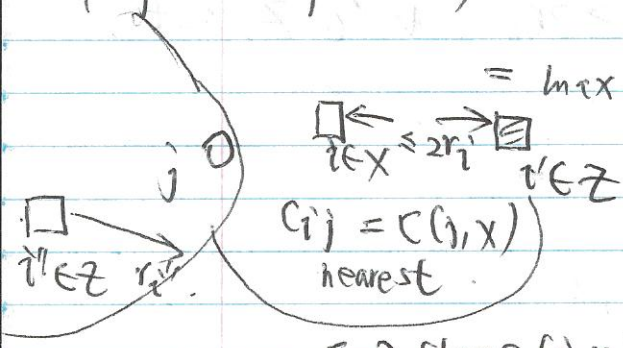
proof: If $i \in Z$, take $i' = i$; otherwise, by defn. of the algorithm i is not picked in step 3), because $\exists i' \leq i$, $i' \in Z$, st. $c_{ii'} \leq 2r_i$.

Fact 4: For any soln. $X \neq \emptyset$, if $i \in X$ is st. $c_{ij} = c(j, X)$ then $\text{charge}(j, X) \geq \max(c_{ij}, r_i)$

proof: $\text{Charge}(j, X) \geq c_{ij} + \max(0, r_i - c_{ij})$

$= \max(c_{ij}, r_i)$

several cases.



$\leq 3 \cdot \text{Charge}(j, X)$. verify

lemma 5: $\forall i \in Z$, $\text{charge}(j, Z) \leq \max(c_{ij}, r_i)$

Fix any soln. $X \neq \emptyset$, any client $j \in C$.

~~charge~~ let $i \in X$ be st. $c_{ij} = c(j, X)$,

$i' \in Z$ be st $i' \leq i$, $c_{ii'} \leq 2r_i$ (exists by lemma 3)

$\Rightarrow \text{charge}(j, Z) \leq \max(c_{i'j}, r_{i'})$ (by lemma 5)

$\leq \max(c_{ij} + c_{ii'}, r_i)$

$\leq \max(c_{ij} + 2r_i, r_i) \leq 3\max(c_{ij}, r_i) \leq 3\,\text{charge}(j, X)$

$v_i((x_i, f_i); X) = \begin{cases} -f_i & ; \text{if } x \in X \\ 0 & ; \text{o/w} \end{cases}$ Check assignment prob. set for non-single dim.

Scheduling problem:

n jobs, m machines. Each job j has a processing requirement of $P_j \geq 0$ work-units. Each machine i has a speed s_i

$= \#$ of work units unit of time ≥ 0

$\Rightarrow$ m/c i takes p_j / s_i time units to process j.

Goal: Assign each job j to a m/c $i(j) \in \{1, \dots n\}$ so as to minimize max. completion time of a job.

ie. $T_{max} = \max_i \left(\sum_{j : i(j) = i} p_j \right) / s_i$ $\rightarrow L_i$ = load on i.

T_{max}: makespan of a schedule / assignment

Mechanism design setting: Each m/c i is a player whose private information is its speed. so

$A = \{$ all possible schedules $s \}$

$V_i = \{ v_i \in \mathbb{R}^A : v_i = -t_i \alpha_i \}$ where $t_i \geq 0$

$\boxed{p_j\text{'s common knowledge}}$

$\alpha_i(s) =$ load on i under $s = \sum_{j : i(j) = i} p_j$

$t_i = \frac{1}{s_i}$ (# time units) / work unit

July 24th

Scheduling M players (m/c's) n jobs job j has p_j processing requirement.

$A = \{$ all schedules $\{ i(j) \}_{j=1}^{n} \}$

$V_i = \{ v_i \in \mathbb{R}^A : v_i = -t_i \alpha_i \}$ where $\alpha_i(s) = \sum_{j : i(j) = i} p_j$

Goal: minimize $T_{max}(s) = \max_i \left(\sum_{j : i(j) = i} p_j \right) t_i$

Monotone approx. algorithm $\equiv$

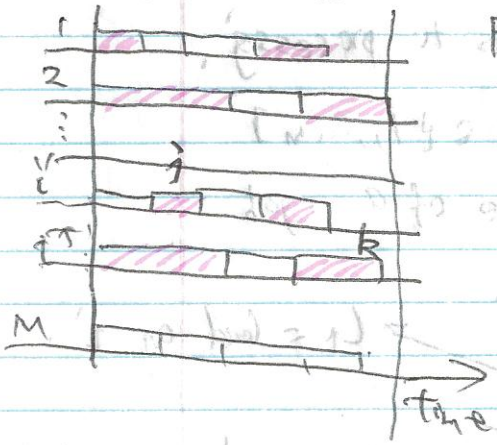
if $s = g(t, t_{-i})$ and $s' = g(t', t_{-i})$ for $t' < t$

then $\underbrace{\sum_{j : i_s(j) = i} p_j}_{\alpha_i(s)} \leq \sum_{j : i_{s'}(j) = i} p_j = \alpha_i(s')$

$OPT := \min_{\text{schedules } s} T_{max}(s)$

lower bound on OPT. Order jobs st. $p_1 \geq p_2 \geq \dots \geq p_n$

order machines st. $t_1 \leq t_2 \leq \dots t_M$



Fix some job j and let i^* be the last M/c on which some job $\{1, \dots j\}$ is scheduled.

Then 1) $OPT \geq p_k t_{i^*} \geq p_j \cdot t_{i^*}$

2) $$\sum_{i=1}^{i^*} \frac{OPT}{t_i} \geq \sum_{l=1}^{j} p_l$$

$$\equiv OPT \geq \left(\sum_{l=1}^{j} p_l\right) \Big/ \left(\sum_{i'=1}^{i^*} 1/t_{i'}\right)$$

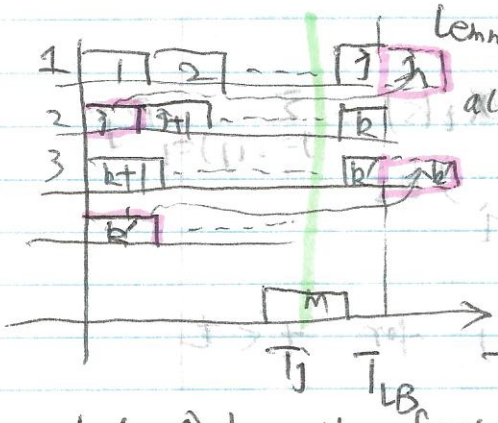
So $$OPT \geq \max\left\{\frac{p_j t_{i^*}}{\cancel{t_{i^*}}}, \frac{\left(\sum_{l=1}^{j} p_l\right)}{\sum_{i'=1}^{i^*} 1/t_{i'}}\right\} \geq \underbrace{\left(\min_{i} \max\left\{\frac{p_j t_i}{\cancel{t_i}}, \frac{\sum_{l=1}^{j} p_l}{\sum_{i'=1}^{i} 1/t_{i'}}\right\}\right)}_{T_j}$$

holds for every j.

$$OPT \geq \underset{\text{lower bound}}{T_{LB}} = \max_{j=1,\dots n} \min_{i=1,\dots m} \max\left\{\frac{p_j t_i}{\cancel{t_i}}, \frac{\sum_{l=1}^{j} p_l}{\sum_{i'=1}^{i} 1/t_i}\right\}$$

A Fractional Schedule:

Compute T_{LB}. and assign jobs "greedily" to m/c's.



Lemma 1: There is enough space to schedule all jobs. Moreover, if j is assigned to M/c i, then

$$T_{LB} \geq \frac{p_j \cdot t_i}{\cancel{t_i}}$$

proof: Suppose there isn't enough space

let j be the first job that we could not assign fully.

$$\sum_{i'=1}^{M} T_{LB}/t_{i'} < \sum_{l=1}^{j} p_l \quad \text{ie. } T_{LB} < \frac{\sum_{l=1}^{j} p_l}{\sum_{i'=1}^{M} 1/t_{i'}}$$

$$= \min_{i}\left(\frac{\sum_{l=1}^{j} p_l}{\sum_{i'=1}^{i} 1/t_{i'}}\right)$$

contradicts defn. of T_{LB}.

let i^* be the index st.

$$T_j = \max\left\{ \frac{p_j}{t_{i^*}}, \frac{\sum_{l=1}^{j} p_l}{\sum_{i'=1}^{i^*} 1/t_{i'}^*} \right\}$$

$T_{LB} \geq T_j$ and since there is enough space on machines $1, \dots i^*$ to process jobs $\{1, \dots j\}$ if j is assigned to i in greedy schedule, then $i \leq i^* \Rightarrow$

$$T_{LB} \geq T_j \geq p_j / t_{i^*} \geq p_j / t_i = p_j t_i$$

$$OPT \geq T_{LB} = \max_j \min_i \max\left\{ p_j t_i, \left(\sum_{l=1}^{j} p_l\right) / \sum_{i'=1}^{i} 1/t_{i'} \right\}$$

Lemma 2: The fractional algorithm is monotone.

proof: Fix i, fix t_{-i}, let $t = (t_i, t_{-i})$

$\tilde{t} = (\tilde{t}_i, t_{-i})$ where $\tilde{t}_i = B t_i$, $B < 1$.

let $S \leftarrow$ schedule for t with load on $i = L_i$

$\tilde{S} \leftarrow$ schedule for $\tilde{t}$. $= \tilde{L}_i$

To show $\tilde{L}_i \geq L_i$

claim: $B T_{LB} \leq \tilde{T}_{LB} \leq T_{LB}$. (simply because $\forall i'$

$$t_{i'} B \leq \tilde{t}_{i'} \leq t_{i'}$$

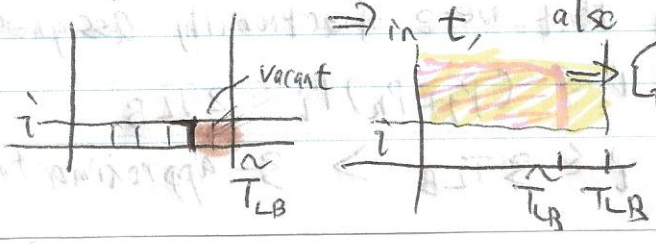
$$\Rightarrow \forall j, \forall i' \quad B \cdot \max\left\{ p_j t_{i'}, \frac{\sum_{l=1}^{j} p_l}{\sum_{i''=1}^{i'} 1/t_{i''}} \right\}$$

$$\leq \max\left\{ p_j \tilde{t}_{i'}, \frac{\sum_{l=1}^{j} p_l}{\sum_{i''=1}^{i'} 1/\tilde{t}_{i'}} \right\} \leq \text{expression for } t.$$

2 cases:

- $\tilde{L}_i = \tilde{T}_{LB} \cdot \tilde{t}_i \geq B\, T_{LB} \cdot \tilde{t}_i = T_{LB} \cdot t_i \geq L_i$
- $\tilde{L}_i < \tilde{T}_{LB} \cdot \tilde{t}_i \Rightarrow$ in $\tilde{t}$, i is the slowest m/c.

$\Rightarrow$ in t, also i is the slowest m/c

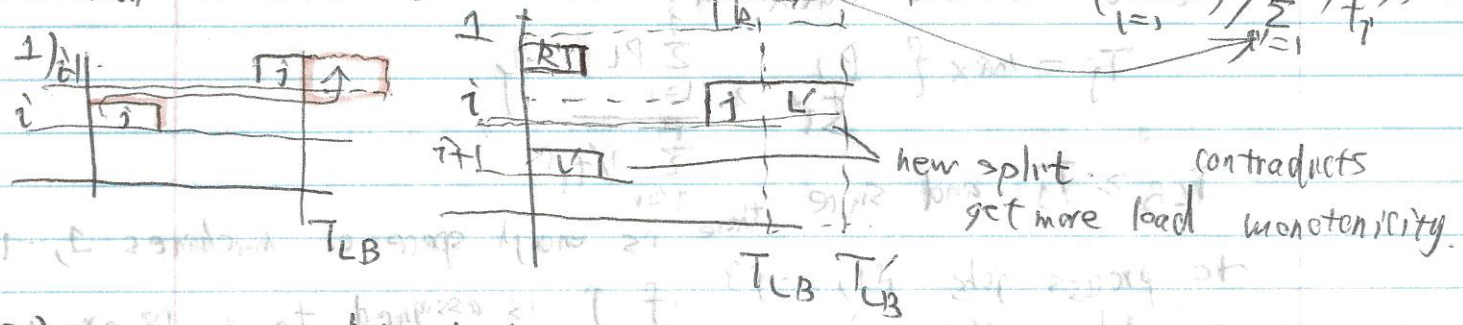


$\Rightarrow \tilde{L}_i = \sum_{j=1}^{m} p_j$ − (loads on other m/c in $\tilde{S}$)

$L_i = \sum_{j=1}^{m} p_j$ − (loads on other m/c's in S)

$\Rightarrow \tilde{l}_i \geq l_i$. Suppose $T_{LB} = \max_{j} \min_{i} \max\{p_j t_i, (\sum_{l=1}^{j} p_l) / \sum_{i=1}^{i} 1/t_i\}$

Conversion to an actual schedule.



2) Randomized "Rounding":

algorithm outputs a random schedule.

Defn. Randomized mechanism $M^R = (f^R, \{P_i^R\})$ where f^R is a randomized algorithm that outputs a random alternative and $P_i^R(V)$ is a random variable.

$\Rightarrow$ utility $(\bar{v}_i; (v_i, v_{-i}))$ is also a random variable.

Truthfulness $\equiv$ each player maximizes his <u>expected</u> utility by declaring true input.

$\forall i, \forall \bar{v}_i, \forall v_{-i}$,

$$E[\text{utility}(\bar{v}_i; (\bar{v}_i, v_{-i})] \geq E[\text{utility}(\bar{v}_i; (v_i, v_{-i})] \quad \forall v_i \in V_i$$

Truthfulness in expectation.

Let x_{ij} = fraction of job j assigned to m/c i $\in [0,1]$ in fractional schedule

with prob x_{ij} assign $j \to i$.

$E[L_i] = \sum_j p_j \cdot x_{ij}$ = load on i in fractional schedule.

$\Rightarrow E[L_i] \uparrow$ if $t_i \downarrow$.

Approximation: each job in the worst case, each m/c i gets at most 2 jobs (j, k) completely that were fractionally assigned to it.

$\Rightarrow$ total extra ~~load~~ time on $i \leq (p_j + p_k) t_i \leq 2 T_{LB}$

$\Rightarrow$ total time taken on $i \leq 3 T_{LB} \Rightarrow$ 3-approximation.

Next lecture July 31 th, 12:20 PM In Rm. MC 5136B

July 29th, CO759. Algorithmic Game Theory.

Combinatorial Auctions: (Multi dimensional Mechanism Design)

n players, set $U = \{1, \cdots m\}$ non-identical items.

Want to model players' having preferences for sets of items.

In particular, want to capture items being

- Complements: Value for a set > Sum of values of items in set.
- SUBSTITUTES: Value for a set < Sum of values of items in set.

Each player i has a set-function $\bar{v}_i : 2^V \to \mathbb{R}$.

where $\bar{v}_i(S)$ denotes i's value for set S of items. and we impose that 1) $\bar{v}_i(S) \le \bar{v}_i(T)$ $\forall S \subseteq T$. monotonicity ~~or free~~ disposal

2) $\bar{v}_i(\emptyset) = 0$ → "Normalization".

Goal: Find an allocation $(S_1, \cdots S_n)$ of items to players that maximize (where $S_i \cap S_j = \emptyset$ $\forall i \ne j$) $\sum_i \bar{v}_i(S_i)$

Mechanism-Design setup: $A = \{$all allocations $(S_1, \cdots S_n)$ st. $S_i \cap S_j = \emptyset$ $\forall i \ne j$

A set-function: $\bar{v}_i$ extends to a vector $\tilde{v}_i \in \mathbb{R}^A$ where

$\tilde{v}_i(a = (S_1, \cdots S_n)) = \bar{v}_i(S_i)$

$V_i = \{$all vectors in $\mathbb{R}^A$ obtained from set-functions v_i satisfying (1) monotonicity & (2) normalization $\}$

$$\sum_{i=1}^{n} \tilde{v}_i(a) = \sum_{i=1}^{n} \bar{v}_i(S_i)$$

Complexity issues:

1) Input Complexity: specifying $\bar{v}_i : 2^V \to \mathbb{R}$ explicitly takes 2^m space → huge input size.

2) Computational Complexity: Even if we consider succinctly describable valuations, called "short valuations", SWM problem is NP-hard.

E.g. suppose each i wants just one set S_i

i.e. $\bar{v}_i(T) = \begin{cases} \bar{v}_i & \text{if } T \supseteq S_i \\ 0 & \text{o/w} \end{cases}$ — single-minded valuations.

such $\bar{v}_i$'s can be described compactly as $(\bar{v}_i, S_i)$ but SWM problem is NP-hard, and cannot be approximated to a factor better than $O(\sqrt{m})$

3) Strategic "Complexity" There is no "convenient" Charatorization of implementable algorithms in general.

Truthful Mechanisms for CAs.

1) Single-minded (SM) valuations/bidders/players.

Each player i's valuation specified by tuple (v_i, S_i) — v_i: not known; S_i: common knowledge

"known" SM case: The sets S_i desired by players are common knowledge

$\hookrightarrow$ becomes single-dimensional setting. Implementability $\equiv$ Monotonicity i.e. if i wins with bid v_i, he also wins with bid $v_i' > v_i$.

Order $(f_1 \cdots f_n)$

Ordering-based algorithms: Given functions: $f_1 : \mathbb{R} \to \mathbb{R}$, $f_2 : \mathbb{R} \to \mathbb{R}$, $\cdots f_n : \mathbb{R} \to \mathbb{R}$.

1) order the players so that $f_1(v_1) \geq f_2(v_2) \geq \cdots \geq f_n(v_n)$ (breaking ties in a fixed way)

2) Initialize W (set of winners) $\leftarrow \emptyset$, $N \leftarrow \{1, \cdots n\}$

3) while $N \neq \emptyset$ [will maitain invariant that $\forall$ players i in N $\forall j \in W$, $S_i \cap S_j = \emptyset$]

- Let i be first player in N.
- $W \leftarrow W \cup \{i\}$, remove from N all players k. st. $S_k \cap S_i \neq \emptyset$.

4) Return $\{S_i\}_{i \in W}$ as allocation.

Theorem: If all f_i's are $\uparrow$ functions then. Exercise

Order $(f_1 \cdots f_n)$ is a monotonic algorithm.

Examples: (i) $f_i(x) = x$, ordering players by $\downarrow v_i$.

Bad example: player 1 wants $S_1 = U$, has value $v_1 = 1+\varepsilon$,

player $2, \cdots m+1$ wants set $S_i = \{i-1\}$ has value $v_i = 1$.

Algorithm chooses allocation $(S_1) \to$ value $= 1+\varepsilon$.

OPT.. chooses $(S_2, \cdots S_{m+1}) \to$ value $= m$

(ii) $f_i(x) = \frac{x}{|S_i|}$ i.e. order players in $\downarrow$ order of $\frac{v_i}{|S_i|}$

Player 1 wants $S_1 = U$ has value $v_1 = m-\varepsilon$

Player $2, \cdots m+1$. want $S_i = \{1\}$, value $v_i = 1$

algorithm chooses e.g. (S_2) as allocation $\to$ value $= 1$

OPT ------ (S_1) ------ $\to$ value $= m-\varepsilon$

(iii) $f_i(x) = \frac{x}{\sqrt{|S_i|}}$ ie. order in $\downarrow$ order of $\frac{v_i}{\sqrt{|S_i|}}$

Theorem: with above f_i's, Order(·) is a $\sqrt{n}$ - approx. algorithm.

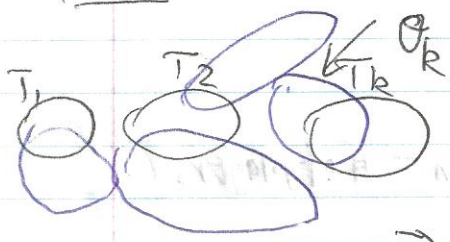
proof: let $(T_1, \cdots T_k)$ be the allocation returned by algorithm.

$(O_1, \cdots O_l)$ be an optimal allocation.

Notation: Each T_i is some set S_p. let $v_{T_i} = v_p$

O_j ------ S_q, let $v_{O_j} = v_q$.

For each T_i, define $\mathcal{O}_i = \{ O_j : O_j \cap T_i \neq \emptyset$ and st. T_i is the first set in the ordering that intersects $O_j \}$



will show that

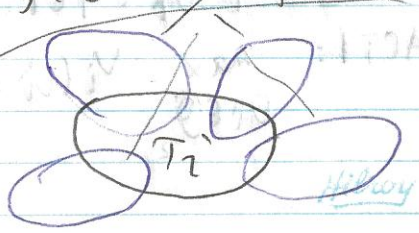
$$v_{T_i} \geq \frac{1}{\sqrt{m}} \sum_{j \in \mathcal{O}_i} v_{O_j} \quad (*)$$

disjoint

$x \notin \mathcal{O}_k$ $\Rightarrow$ adding $(*)$ $i = 1, \cdots k$ and since the $\mathcal{O}_i$s partition $\{O_1, \cdots O_l\}$ we get

$$\sum_{i=1}^{k} v_{T_i} \geq \frac{1}{\sqrt{m}} OPT.$$

Disjoint



Fix some i, RHS of $(*)$ is

$$\sum_{j \in \mathcal{O}_i} v_{O_j} = \sum_{j \in \mathcal{O}_i} \frac{v_{O_j}}{\sqrt{|O_j|}} \sqrt{|O_j|}$$

$$\leq \frac{V_{T_i}}{\sqrt{|T_i|}} \left(\sum_{j \in O_i} \sqrt{|O_j|} \right)$$

Cauchy-Schwarz inequality

$$\sum_{j=1}^{p} a_j b_j \leq \left(\sum_{j=1}^{p} a_j^2 \right)^{1/2} \cdot \left(\sum_{j=1}^{p} b_j^2 \right)^{1/2}$$

Take $p = |O_i|$ $a_j = 1$ $b_j = \sqrt{|O_j|}$ $\forall j = 1, \dots p$

$$\leq \frac{V_{T_i}}{\sqrt{|T_i|}} \cdot \underbrace{\left(|O_i| \right)}_{\leq |T_i|}^{1/2} \cdot \underbrace{\left(\sum_{j=1}^{|O_i|} |O_j| \right)}_{\leq M}^{1/2}$$

$$\leq V_{T_i} \cdot \sqrt{m}$$

CO759 Algorithmic Game Theory. July 31 2008.

Single-minded valuations algorithm from last class

1) $\forall i$, $\forall V_{-i}$ $\forall S_i$

$\exists$ threshold $t_i(S_i)$

2) If $S \subseteq T$, then $t_i(S) \leq t_i(T)$ $\forall i$

CAS with subadditive valuations

Every player i's $\bar{V}_i$ is st.

$$\bar{V}_i(S \cup T) \leq \bar{V}_i(S) + \bar{V}_i(T) \quad \forall, S, T.$$

"substitutes" property

Approximation Algorithm:

Input: $(V_1 \dots V_n)$

Let $(O_1, \dots O_n)$ be optimal allocation.

Let $\mathcal{O}_1 = \{ O_i : |O_i| \leq \sqrt{m} \}$ m = # of items. n = # of players

Small group

$|\mathcal{O}_2| \leq \sqrt{m}$

$\mathcal{O}_2 = \{ O_i : |O_i| > \sqrt{m} \}$ — large group

FACT 1: $$\max_{O_i \in \mathcal{O}_2} V_i(O_i) \geq \frac{1}{\sqrt{m}} \cdot \underbrace{\sum_{O_i \in \mathcal{O}_2} V_i(O_i)}_{OPT_2}$$

NOTATION: $OPT = OPT_1 + OPT_2$ where $OPT_1 = \sum_{O_i \in \mathcal{O}_1} v_i(O_i)$

$OPT_2 = \sum_{O_i \in \mathcal{O}_2} v_i(O_i)$

$$\Rightarrow \max_{i=1,\dots,n} v_i(U) \geq \max_{i: O_i \in \mathcal{O}_2} v_i(O_i) \geq \frac{1}{\sqrt{m}} OPT_2 \quad \dots (a)$$

FACT 2: For every $O_i \in \mathcal{O}_1$

$$\max_{e \in O_i} v_i(e) \geq \frac{v_i(O_i)}{\sqrt{m}}$$

$\{ \forall i, \forall S, \; v_i(S) \leq \sum_{e \in S} v_i(e) \leq |S| \max_{e \in S} v_i(e)$

Consider the allocation where we give each i s.t. $O_i \in \mathcal{O}_1$ the item $e_i^* = \arg\max_{e \in O_i} v_i(e)$. Then we get an allocation of total value $\sum_{i: O_i \in \mathcal{O}_1} v_i(e_i^*) \geq \sum_{i: O_i \in \mathcal{O}_1} \frac{v_i(O_i)}{\sqrt{m}} = \frac{OPT_1}{\sqrt{m}} \dots (b)$

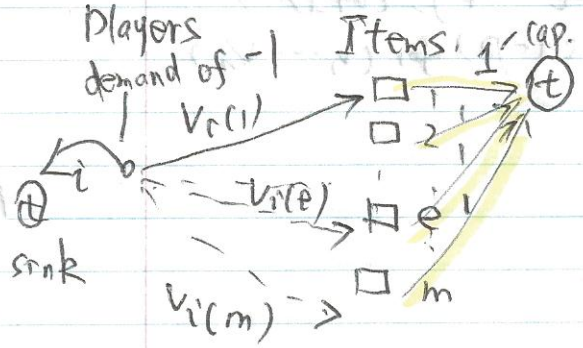
Consider the algorithm where we take the better of the following 2 allocations

(A) Let i^* be s.t. $v_{i^*}(U) \geq v_i(U) \; \forall i = 1, \dots m$. Assign i^* all items.

(B) Assign each player i at most one distinct item e_i so as to maximize $\sum_i v_i(e_i)$

Theorem: Above algorithm returns an allocation of total value at least

$$\geq \max\left(\frac{OPT_1}{\sqrt{m}}, \frac{OPT_2}{\sqrt{m}} \right) \geq OPT / 2\sqrt{m}.$$



max flow. efficiently implement (B) (in polynomial time)

Note that algorithm defines the SCF g where

$$g(v) = \arg\max_{a \in A'} \sum_i v_i(a)$$ where $A' = \{$all allocations that assign all items to 1 player$\} \cup \{$allocations that assign ≤ 1 item to each player$\}$

SWM problem over a restricted alternative set.

But VCG result still applies $\Rightarrow$ can use VCG prices to get a truthful mechanism.

Linear program $\quad$ LP-based c-approx. algorithm.

$$(CA\text{-}P) \quad \max \sum_{i,S} v_i(S) X_{i,S}$$

$$\text{s.t.} \quad \forall i \quad \sum_S x_{i,S} \leq 1$$

$$\sum_i \sum_{S: j \in S} x_{i,S} \leq 1$$

$$\forall i, S \quad x_{i,S} \geq 0.$$

Fractional Mechanism Design Problem.

~ Compute a fractional allocation, ie, soln. to $(CA\text{-}P)$ that maximizes SW.

$\mathcal{P}$ = set of all feasible solns to $(CA\text{-}P)$

ie, alternative set is $A' = \mathcal{P}$.

Player i with set functions.

$v_i : 2^V \to \mathbb{R}$. assigns the value

$$v_i(x) = \sum v_i(S) x_{i,S}$$ to an alternative $x \in \mathcal{P}$.

Step 1: Construct fractional mechanism $\mathcal{M}^F = (g, \{p_i\})$

where $g(v) = x^*(v)$ = optimal soln. to $(CA\text{-}P)$ for $(v_1 \cdots v_n)$

$\{p_i(v)\}$ = VCG prices

$\mathcal{M}^F(g, \{p_i\}) \to \mathcal{M}^R = (h, \{r_i\})$

1) $x^*(v) \to \{y_1 \cdots y_L\}$ w.p. $\lambda_1 \cdots \lambda_L$.

$\to$ integer pts in $\mathcal{P}$.

2) $\{P_i(v)\} \to \{r_i(v)\}$

will ensure that

$$\mathbb{E}[\text{utility}_i^{MR}(\bar{v}_i(v_{-i}))] = \frac{1}{c} \cdot \text{utility}_i^{MF}(\bar{v}_i, (v_i, v_{-i})) \quad (*)$$

Decomposition Lemma : based on [Carr-Vempala 02]

will use A to get $\{(\lambda_k, y_k)\}$ st.

$$\frac{x^*}{c} = \sum_{k=1}^{R} \lambda_k y_k .$$

c scale.

lies on the convex hull of Extreme pts.

$$\mathbb{E}[r_i(v)] = \frac{P_i(v)}{c}$$

proof of truthfulness. packing property.

Goal: $\mathcal{P}_I = \{y \in \mathcal{P} : y_{i,s} \in \{0,1\} \ \forall i, s\}$

Find $\{\lambda^{(y)}\}_{y \in \mathcal{P}_I}$ st.

$$\sum_{y \in \mathcal{P}_I} \lambda^{(y)} \cdot y = x^*/c \qquad \lambda^{(y)} \geq 0, \quad \forall y \in \mathcal{P}_I .$$

$$\sum_{y \in \mathcal{P}_I} \lambda^{(y)} = 1.$$

$E = \{(i,s) : x^*_{i,s} \underset{\text{strict}}{>} 0\}$

(primal) $\min \sum_{y \in \mathcal{P}_I} \lambda^{(y)}$

st. $\sum_{y \in \mathcal{P}_I} \lambda^{(y)} \geq x^*_{i,s}/c \ ] \times w_{i,s}$

$\sum_{y \in \mathcal{P}_I} \lambda^{(y)} \geq 1 \ ] \times \zeta$

$\lambda^{(y)} \geq 0 \quad \forall y \in \mathcal{P}_I$.

(Dual $\max \sum_{(i,s) \in E} \frac{w_{i,s}}{c} + \zeta$

st. $\forall y \in \mathcal{P}_I \ \sum_{(i,s) \in E} y_{i,s} w_{i,s} + \zeta \leq 1$

$\forall (i,s) \in E \ w_{i,s}, \zeta \geq 0$

Think of $w = (w_{i,s})_{i,s}$ as specifying a vector of valuation functions

Dual constraint: Every integer soln. has sw- value under w of at most $1-\zeta$.

$\rightarrow$ implies $OPT_{(D)} \leq 1$ (since $\sum_{i,s} w_{i,s} x_{i,s} \leq C$ (max value of an integer soln $y \in P_I$)

$\Rightarrow OPT_{(D)} = OPT_{(P)} = 1$

Moreover, can determine in polytime using A if (w, z) is feasible to (D) $\Rightarrow$ can solve (D) (P) in poly time.

End of the Course $\boxtimes$